my Zaurus SL-C3000 and SL-C3100

Sharp Zaurus SL-C3000 and SL-C3100

© Copyright 2006 Hd Luc (hdluc@yahoo.com)

I got the Zaurus SL-C3000 and SL-C3100. These impressive little devices are almost perfect. They are made by Sharp, however, Sharp decided to only sell these models to the Japanese market.

Being in Australia this was a big disappointment. I tried to order one from Conics, but got no response from them for several weeks, which I think is unacceptable, so I asked a friend of mine who lives in Tokyo to send me one. It's so nice to have friends all over the world :)

I want to thank Sachiko for getting me my first Zaurus. Thank You very much Sachiko!!! You are the best!

The Zaurus is my replacement for my mini Laptop, the **Toshiba Libretto 50 CT**, which is now hosting this website running Redhat Linux.

The Sharp Zaurus SL-C3000 is the first step of my dream laptop becoming reality. I was sick of carrying a bulky and heavy laptop around and the other PDAs on the market did not appeal to me except for Sony's little Clie but that is another story. After the SL-C3000 died, I bought my second Zaurus, the SL-C3100 from PriceJapan. I ordered it Friday afternoon and it took 3 days to arrive in Sydney from Tokyo and clear customs. However, the local Post Office fucked up as usual and it took 2 more days to arrive. The Zaurus is an astounding technology gadget dreamt up by the Japanese and is the only PDA that really suits my needs. For instance, it runs **Linux**! Also the SL-C3000 was the first PDA with an internal HDD, and currently the SL-C3000 and SL-C3100 are one of the very few PDAs with an internal HDD. The 4 GB micro drive offers sufficient storage (for a PDA) and opens up many possibilities, such as using the Zaurus as a portable video and audio player. The Zaurus' HDD is also recognized by Windows as a plug-n-play USB storage device (but sometimes it is a pain to detect properly). The display of the Zaurus SL-C3000 and SL-C3100 is 3.7" in size and the VGA (640x480) screen is truly awesome. It's bright and razor sharp. And there's a built-in zoom function that allows you to zoom the screen. The screen's orientation automatically adjusts when swivelled. The swivelling screen transforms the shape of the Zaurus from PDA-style to laptop-style. Once in laptop-style, you can utilize the QWERTY keyboard, with its great tactile feedback. The Zaurus features an advanced character recognision for Japanese (kanji, hiragana and katakana) and English which is entered directly onto the touchscreen via a stylus pen. It also has a fully featured bi-directional Japanese-English dictionary and translation software. The Zaurus measures 124x87x25mm, and weighs 298g.

And here is my Zaurus next to my new PC. It's tiny in comparison.

Specs:

Features:

- Intel XScale PXA270 CPU at 416MHz
- 640x480 transreflective touch screen, 3.7" diagonal, backlit
- 4 GB internal hard drive
- 64 MB memory (SDRAM)
- 16 MB Internal Flash for SL-C3000 / 128 MB Internal Flash for SL-C3100
- USB host capability (USB OTG implementation)
- Weight 298g, Size 128x87x24 mm
- 1800 mA Lithium Battery (up to 7 hours)
- White Colour (SL-C3000) / Black Colour (SL-C3100)
- Compact Flash (CF-II) and Secure Digital (SD) expansion slots
- Infra-Red port
- 3.5mm stereo audio out/microphone in, plus small built-in speaker
- Landscape/Clamshell or portrait style, hot-switchable
- Full QWERTY keyboard with bonus Japanese characters
- Operating System: Linux kernel 2.4.20 from Lineo
- Graphical User Interface (GUI): Qtopia 1.5.4 from Trolltech
- Suite of Personal Information Management (PIM) applications
- Hancom Mobile Office
- Broad compatibility with established Sharp Zaurus software base

USB OTG:

The SL-C3000 and SL-C3100 use the new USB OTG (On-The-Go) connector which allows both mini A and mini B plugs to be plugged in. When a mini B is plugged in, the Zaurus will act as a USB client, ie the Zaurus becomes a USB storage device to the connected host PC or Laptop. When the mini A is plugged in, the Zaurus will act as a USB host and you can attach USB devices such as keyboards, mouse, disks, etc. to it.

Here is how USB mini A and USB mini B look like side by side:

The standards for USB are defined by the USB-IF (USB - Implementers Forum). The OTG 1.0 is a new extension to the USB 2.0 specifications. USB 2.0 defines the following speeds:

- USB 2.0 High Speed - 480Mbps
- USB 2.0 Full Speed - 12Mbps (same as USB 1.1 speed)
- USB 2.0 Low Speed - 1.5Mbps

In order to be USB 2.0 compliant, one or more of the above speeds has to be implemented. The Zaurus implements the USB 2.0 Full Speed data rate for file transfers and the USB 2.0 Low Speed data rate for peripherals such as mice and keyboards. It does not implement the USB 2.0 High Speed data rate.

C3000 vs. C3100:

The most noticable difference between the C3000 and the C3100 is their colour. The C3000 has a smooth white cover whereas the C3100 is mainly black with dark silver frame for the screen. The C3100 case also feels like it is made of granite while the C3000 feels more metallic and smooth.

Internally, the difference is the amount of flash memory. The C3000 only has 16MB whereas the C3100 has 128MB. This difference in flash memory does give the C3100 a small boost in speed over the C3000 (no spin-up time when loading applications).

The software works the same in both models, but there is some differences in management and maintenance when going down to the details level. There is also more software provided with the C3100, but the additional software is all in pure Japanese only. The C3000 comes with 2 CD-ROMs whereas the C3100 comes with 3 CD-ROMs, the additional CD contains data for the extra software.

The other difference is that the C3000 came with a tiny power adaptor which only worked with 100V input (Japan only). The C3100 comes with a slightly bigger power adaptor which works with 100-240V input.

Accessories:

Although the **Zaurus SL-C3000** and **Zaurus SL-C3100** are great little devices, there are a few essential things missing which I had to add in order to make them perfect. Since I live in Australia, the **power adaptor** was a little problem with the SL-C3000 since it only came with a 100V power adaptor and I needed to find a compatible adaptor. Both Dicksmith and Tandy have a relatively small adaptor (made by Digitor) which has the required 5V input at 1A for around AUD $40. Jaycar even has one for AUD $30 (another brand and

slightly bigger). The SL-C3100 comes with a 100-240V power adaptor (5V input at 2A) so all that was needed was a cheap Australian plug adaptor.

In addition, I wanted to have wireless networking so I can surf the net, use Yahoo Messenger and IRC as well as share files with my other computers over my home network. This requires a wireless CF adaptor card, but getting one was not such an easy task since Australia is quite behind Asia in terms of the technology market. Most computer shops did not even know what a wireless CF card was and those who knew had to order it in (3-4 days) since there was no demand for them and hence none of them had any in stock. The Zaurus supports 802.11b wireless compact flash and generally in Australia this means the Netgear MA701 or the D-LINK DCF-660W Air. I got myself the **Netgear MA701** which costs around AUD $100 because none of the shops had a D-Link in stock. I recently also found a SanDisk Connect Plus 128M for only AUD $79.

I also found a **USB mini A host adaptor cable** to enable the USB host capability of the Zaurus. Harris Technology has them (**GoldX 5-in-1**) for around AUD $40 and Office Works has them (Comsol 5-in-1) for only AUD $30. The cable comes with a little leather pouch which contains five different interchangable USB connectors: USB Male A, USB A Female, USB B Male, Mini USB A and Mini USB B. I also found some cheaper cables made by Avico. The **Avico USB Digital Camera Cord** comes in several models. The CC1542 is a USB A to USB mini A cable and the CC1522 is a USB mini A to USB mini B cable and costs AUD $9.95 each.

I also wanted to connect my Zaurus to different LANs via ethernet networks and got myself a tiny **USB LAN adaptor**. This cost me AUD $30. In addition to that, I have a small **retractable CAT5 network cable** made by Aidata (bought from Japan on my last trip there for approximately AUD $20).

Since a single USB connection is rather limiting, I also got a **Blue Eye powered mini USB Hub** so I can connect up multiple USB devices at the same time. I opted for a powered hub instead of a cheaper unpowered one which I had lying around. With a powered hub I can connect my **BlueEye 40 GB 2.5" USB HDDs** to the Zaurus. The 5V at 1A power required by the HDD had to come from the USB Hub. But be careful, most powered USB hubs only provide 5V at 500mA which is not sufficient for some harddisks. The Blue Eye hub costs around AUD $40 and is one of the few that supplies the necessary 1A. In fact, its AC adaptor supplies 2A, which means it can handle more than 1 harddisk. Most 2.5" disks only need 500mA to run, but 1A to power up. 3.5" harddisk enclosures normally come with an AC power so those can be used with an unpowered hub and I was able to use my **BlueEye 200 GB 3.5" USB HDD** which was NTFS formatted with the Z. I have also tried the BlueEye hub with four disks plugged in without problems. While shopping around I have found another place which sells the Blue Eye Hub for only AUD $25, so I got myself another spare. Also, the USB hub powers my USB desklight which costs around AUD $20. BlueEye devices are made/distributed by Noontech.

The hub also allowed me to simultaneously use my **mini USB mouse** and **mini USB keyboard**. The mouse is particular useful when running X windows on the Z. The keyboard I normally only use with my Zaurus when I need to type a lot. Almost any standard USB keyboard should work with the Zaurus. The keymapping needs some adjustments since the Zaurus keymapping layout is Japanese and not English by default but this can be easily changed since it is mainly QWERTY based.

I also got a pack of spare stylus so I can leave one at the office. In addition, I bought the **Belkin 4-in-1 stylus** for around AUD $39. It is the size of a normal pen so it feels quite natural to use. When you use it, you get the feeling that the stylus tip is very soft and won't damage your screen. The tip can also be retracted like most pen by a simple twist and it even has a pen built-in that can be used to write. On

the other end there is a light and a laser pointer. The laser pointer is quite handy since I do quite a lot of presentations and the light is pretty cool and helps in the dark. An exact copy of the Belkin 4-in-1 can also be found at Jaycar for half the price, and for an even cheaper copy, try Vietnam where you can get it for around AUD $10 only. The copies also come in a wooden pen box with extra spare batteries as well.

Photos from my digital camera can be easily transferred to the Zaurus either via USB connection (using the Avico or GoldX cables) or by directly inserting the SD Memory Card from the camera into the Zaurus' SD slot. Some of my other cameras use CF Memory Cards, but the Zaurus has a slot for them as well. I also got myself a few additional SD and CF cards, in particular, the **4GB Kingston CF** card and the **4GB pqi SD** card. Now I got several SD and CF memory cards to swap in and out, and I also got a Kodak PCMCIA adaptor so I can use the CF cards in my Laptop too. The Zaurus made my All-in-One USB card reader redundant, however, I also bought a 4-in-1 CF adaptor which allows me to use additional memory cards in my Zaurus such as my **2GB memory stick duo**. I also got the new **2GB SanDisk Ultra II plus USB** which is a mini SD card with a USB adaptor stuck on it so you can use it as a normal SD as well as a USB stick.

Finding a **Leather Carrying Case** that fit the C3000/C3100 was another challenge. The normal PDA cases were not made for the C3000/C3100 but I eventually found a leather carrying bag for cameras that was perfect. It even had a little pouch on the front for my CF cards and could be attached to a belt or hung around the neck. The **Tamrac Digital Camera Bag** is made of ballistic nylon and genuine leather. It costs AUD $45

I also bought a **Joytech PSP power extender** and since it has the same plug and power requirements as the Zaurus, I can use it with either my Zaurus or PSP to extend the battery power on a long haul flight. In addition, I bought an **AC to USB adaptor** which has several uses, ie charging my Zaurus, my Mobile phone, or powering my USB hardisk via a **USB y-cable**.

And here is my Zaurus (C3000) in all its glory surrounded with gadgets for it.

Additionally, I had to get the following from Tokyo since I could not get them locally in Australia. Thanks again to Sachiko for going shopping for me.

The **iRiver USB cable** works with the Zaurus. This cable is quite small and has a USB mini A connector on one end and a USB A Female on the other. It will be a handy addition to supplement my GoldX cable. It is currently also the cheapest USB mini A host controller cable available that I could find.

I also found a small USB power cable, the **Diatec P-Cord** (P to Go). This little gem allows me to leave the bulky power adaptor at home. It can power the C3000/C3100 whilst plugged into a USB port of a PC or a Sun Ray (or even my spare USB hub). As a bonus this cable is also retractable. It is perfect for stuffing into a small pocket. In addition, this cable also has a little cousin which also includes a USB mini-b adaptor for synching as well so you can use it to charge and sync at the same time.

The screen on the Zaurus attracts quite a lot of dust. The **OverLay Brilliant** screen protector should protect my little Z's touchscreen. It will hopefully ensure my Z's screen will not be damaged and also reduces the glare when under direct sunlight, so now I can see what's on the screen even outside when the sun is shining on my Z. It can also be easily washed in warm water so cleaning is a breeze, but re-attaching it afterwards is a bit tricky. The inner side easily catches little dust particles which are very hard to remove and they cause bubbles on the screen. However, most importantly, it prevents scratches on the Z's screen.

Well, I already have a few spare styli, but one can never have enough spares, so I got the **Pilot Pentopia stylus**. This stylish replacement stylus looks and feels much better than Sharp's greyish plastic stylus that comes with the C3000 or the black stylus that comes with the C3100. The Pentopia stylus comes in a cool metallic colour with black ends and a red tip. And it is also extensible by a few centimeters to give it the length of a pencil. The extra metal weight also makes it feel like something with substance. It also has a pen under the black cap on the other end.

I have found a USB to VGA adaptor which can be used with the Zaurus. It is made by Kairen and contains a SiS 315E graphics chip and there is a custom driver for it ported to the Zaurus. With the driver and special application installed, you can use this adaptor to connect your Zaurus to a VGA monitor or projector. However, you will need a USB host cable (mini A) and a powered hub. You cannot plug this adaptor straight into the Zaurus. It requires additional power from the USB hub. The USB VGA adaptor was designed for USB high speed (480Mbps) in mind and runs very slow on the Zaurus which does not support this speed.

While in Thailand, I went to the Pantip IT Mall in Bangkok and found a few extra goodies.

The most useful item I believe was the Sony MDR-138 Super Bass Headphones (I don't think Sony really made it but who cares). This set of headphones has a retractable spindle for the cable and a clip for attaching it convenniently while it is retracted. It costs me only 280 Baht which is less than AUD $10 and of much better quality than the usual retractable headphones that are now swamping the computer stores here locally. I also got a few other headphones, some with built-in microphones similar to mobile phone headsets but with two earpieces and 3.5mm connectors, and some of them are also retractable.

I also bought a slim USB combo CDRW/DVD drive for only 3500 Baht which is around AUD $110. They also had battery powered USB harddisk enclosures for 1600 Baht which can be used to self power 2.5" USB harddisks without the need for an AC adaptor or powered hub.

Another bargain I picked up was the Kingston 4 GB Compact Flash memory card which cost me only 9500 Baht, around AUD $300. In addition, I picked up a spare battery for my Z quite cheaply too.

Also found some retractable USB mini B cables for only 100 Baht each (around AUD $3), so I could not resist and bought a few extra spare ones. Unfortunately, they did not have retractable USB mini A cables.

Software:

Included Software:

The Zaurus SL-C3000 and SL-C3100 are LINUX-based PDAs. They come bundled with the following software:

- HancomWord (for .doc),
- HancomSheet (for .xls),
- TextEditor (for .txt),
- ImageViewer (for .jpg, .bmp, .gif),
- MoviePlayer (for .mpg),
- MusicPlayer (for .mp3),
- NetFront Browser,
- E-Mail Client,
- ToDo List,
- Calendar,
- AddressBook,
- Calculator,
- WorldTime,
- Dictionary,
- Translator,
- Terminal Window (need to be installed from CD-ROM),
- Telnet and FTP client (command line),
- English and Japanese handwriting recognition and keyboard input methods.

The C3100 also has the following software pre-installed by default which the C3000 does not have (These applications are only really useful if you are in Japan or know Japanese):

- BunkoViewer
- KiokuDojo (Memory Trainer)
- MobileMap
- Norikae (Transfer Guide)
- Database

The C3000 requires that you update/add the following software wheras the C3100 already comes with the updated versions of these:

- musicplayer2 - [musicplayer-C3000_2.0.0_arm.ipk]
- netfront3.1 - [netfront3.1-C3000_1.5.2_arm.ipk]
- photostorage - [photostorage_1.0.1_arm.ipk]

You should patch the SL-C3000 with the newer 1.11 ROM (card_update_3000111.exe) if it came with the older 1.01 ROM. Also, there are some new drivers and patches that you might need at http://support.ezaurus.com/sl-c3000/update/

The SL-C3100 should come with a 1.02 ROM. New drivers and patches for the SL-C3100 are located at http://support.ezaurus.com/sl-c3100/update/

The SD card driver shipped with the SL-C3000 and SL-C3100 only supports SD cards of sizes up to 1GB. The SL-C3200 comes with an updated SD driver which supports larger SD cards. This driver also works for the SL-C3000 and SL-C3100, however, Sharp only released it pre-installed on the SL-C3200. It has been extracted and re-packaged by the Zaurus user community.

Additional Software:

Even with all this software, there is still a lot of extra functionality that can be obtained by installing additional packages. The following is a list of applications and utilities that I have installed, most of which can be found on the ZUG feed (see Feeds section). I have provided the full filename of the packages which will make locating them with your favourite search engine much simpler. Also, I have archived up all my installed packages here.

Applications and Utilities

for Qtopia GUI

- explorer - [explorer_1.0_arm.ipk]
- filelaunch - [filelaunch_0.4.5_arm.ipk]
- FNViewer - [FNViewer_1.3_arm.ipk]

- FreeNote - [FreeNote_1.13.2pre_arm.ipk]
- guigrep - [guigrep_1.3_arm.ipk]
- htmleditor - [htmleditor_1.0.0_arm.ipk]
- iconedit - [iconedit_0.9.4_arm.ipk]
- jabp - [jabp_1.1.4_arm.ipk]
- java-jed - [java-jed_1.0_arm.ipk]
- java-jportscan - [java-jportscan_1.0_arm.ipk]
- jftp - [jftp_0.23.1_arm.ipk]
- kaddressbook - [kaddressbook_2.1.2_arm.ipk]
- kani - [kani_1.2.0_arm.ipk]
- kino2 - [kino2_0.4.3c_arm.ipk]
- kismet-qt - [kismet-qt_2.0.0-3_arm.ipk]
- konqueror (browser) - [konqueror_snapshot_20020311_arm.ipk]
- kopiemail - [kopiemail_2.1.2_arm.ipk]
- korganizer - [korganizer_2.1.2_arm.ipk]
- minide - [minide_1.2_arm.ipk]
- mioreaderlite - [mioreaderlite_0_08_arm.ipk]
- mooview - [mooview_0.2_arm.ipk]
- neocal - [neocal_1.7.7-2_arm.ipk]
- opera7.3 (browser) - [opera_sl-5x00-7.30.9965_arm.ipk]
- opie-embeddedkonsole - [opie-embeddedkonsole_1.5.9-2_arm.ipk]
- opie-irc - [opie-irc_0.9.1-20020923_arm.ipk]
- opie-keypebble - [opie-keypebble_1.0.0-1_arm.ipk]
- opie-reader - [opie-reader_0.7h_arm.ipk]
- opie-security - [opie-security_1.5.0-20020319_arm.ipk]
- opie-wellenreiter - [opie-wellenreiter_1.0.2.1-20031220_arm.ipk]
- opieftp - [opieftp_0.9.1-20020702_arm.ipk]
- opie-reader - [opie-reader_0.7m_arm.ipk]
- petitepeinture - [petitepeinture_1.4a-1_arm.ipk]
- poqetp - [poqetp_0.0.6_arm.ipk]
- qazoo (yahoo messenger clone) - [qazoo_0.8.1_arm.ipk]
- qflashplayer - [qflashplayer_0.1.0-1_arm.ipk]
- qkonsole - [qkonsole_0.9.3-20040205_arm.ipk]
- qpPhoto - [qpPhoto_1.0.3_arm.ipk]
- qpaint - [qpaint_1.0.0_arm.ipk]
- qpdf2 - [qpdf2_freetype-2.2.1-20040217_arm.ipk]
- qpe-terminal - [qpe-terminal-ja_1.5.0-3_arm.ipk]
- qpe-voicerec - [qpe-voicerec_1.5.0-7_arm.ipk]
- qpzidian - [qpzidian_0.1_arm.ipk]
- qtopia-advancedfm - [qtopia-advancedfm_1.0_arm.ipk]
- shellcommander - [shellcommander_0.6.2-1_arm.ipk]
- SimpleEdit - [simple-edit_1.0.3_arm.ipk]
- TinyClassDeveloper - [TinyClassDeveloper_arm.1_arm.ipk]
- tinyviewer - [tinyviewer_0.3.1_arm.ipk]
- treeexplorer - [treeexplorer_1.7.0-2_arm.ipk]
- visualq - [visualq_0.5.6E_arm.ipk]
- yedit - [yedit_0.1-0_arm.ipk]
- zbedict - [zbedic_0.9.4-0_arm.ipk]
- zeditor - [zeditor_3.3.2english_arm.ipk]
- ziciz (irc) - [ziciz_0.88_arm.ipk]
- zmeeting - [zmeeting_1.1_arm.ipk]
- zplayer - [zplayer_0.0.5_arm.ipk]
- zuc - [zuc_640x480-1_arm.ipk]

for Qtopia Console

- aircrack - [aircrack_2.1-1_arm.ipk]
- gawk - [gawk_3.1.5_arm.ipk]
- bitchx (irc) - [bitchx_1.1-final_armv5tel.ipk]
- bzip2 - [bzip2_1.0.2-1_arm.ipk]
- cpio - [cpio_2.6_arm.ipk]
- dos2unix - [dos2unix_1.0-1_arm.ipk]
- fdisk - [fdisk_2.11g-4_2_arm.ipk]
- file - [file_3.39-2_arm.ipk]
- flite - [flite_arm_bin.tar.gz]

- hping2 - [hping2_2.0.0-rc3-3_arm.ipk]
- imgresiz - [imgresiz_1.0_arm.ipk]
- lftp - [lftp_2.6.7-1_arm.ipk]
- links - [links_2.1pre17-1_arm.ipk]
- lsof - [lsof_4.57-1_arm.ipk]
- madplay - [madplay_0.15.2b_arm.ipk]
- mc - [mc_4.6.0_arm.ipk]
- memcoder_1.1.0-1_arm.ipk]
- mplayer-bvdd - [mplayer-bvdd-iwmmxt_1.1.5-1_arm.ipk]
- ncftp - [ncftp_3.1.5-1_arm.ipk]
- netctl - [netctl_0.3.0-1_arm.ipk]
- nmap - [nmap_3.70-3_arm.ipk]
- openssh-client - [openssh-client_3.6.1p1_arm.ipk]
- qarg - [qarg_0.1.0-1_arm.ipk]
- qcop2 - [qcop2_0.1.0-1_arm.ipk]
- qshdlg - [qshdlg_0.6.2-1_arm.ipk]
- qttkdlg - [qtkdlg_0.4.1-1_arm.ipk]
- shine - [shine_0.01_arm.ipk]
- smbmount - [smbmount_0.1_arm.ipk]
- spxrec - [spxrec_0.0.1_arm.ipk]
- sudo - [sudo_1.6.3p7-2_arm.ipk]
- top - [top_0.3.6_arm.ipk]
- unrar - [unrar_3.5.4-lite-1_arm.ipk]
- unzip - [unzip_5.24_arm.ipk]
- vim - [vim_6.1_arm.ipk]
- wget - [wget_1.10.2_arm.ipk]
- zip - [zip_2.3_arm.ipk]

Services and Runtimes

- apache - [apache-1.3.27-php-4.2.3_0.1_arm.ipk]
- clipboard-applet - [clipboard-applet_1.0.0-1_arm.ipk]
- cool-icons - [cool-icons_0.0.1_arm.ipk]
- dosbox - [dosbox_0.6.3_arm.ipk]
- dummydev - [dummydev_0.01-1_arm.ipk]
- iproute - [iproute_z2.2.4-now-ss991023-1_arm.ipk]
- java 1.3 - [java1.3_1.01-oxy2_arm.ipk]
- jeode - [jeode_1.10.7_arm.ipk]
- joyenabler - [joyenabler_1.3_arm.ipk]
- keyhelper - [keyhelper_1.2.2-1_arm.ipk]
- keyhelperconf - [keyhelperconf_0.3.0-1_arm.ipk]
- kismet - [kismet-2005-08-R1-arm.ipk]
- LUSScreenSaver - [LUSScreenSaver_1.4.6-1_arm.ipk]
- LUSScreenSaverUtil - [LUSScreenSaverUtil_1.3.5-1_arm.ipk]
- LUSSSFish - [LUSSSFish_1.0.0-1_arm.ipk]
- LUSSSMessage - [LUSSSMessage_1.0.0-1_arm.ipk]
- LUSSSPicture - [LUSSSPicture_1.1.1-1_arm.ipk]
- LUSSSUniverse - [LUSSSUniverse_1.0.0-1_arm.ipk]
- nethelper - [nethelper_0.3.0-1_arm.ipk]
- opie-sh - [opie-sh_0.5.1-20020527_arm.ipk]
- Plasterer - [plasterer_2.1.0-1_arm.ipk]
- perl - [perl_5.6.1_arm.ipk]
- qcoptest - [qcoptest_0.1.1_arm.ipk]
- qpe-suspendapplet - [qpe-suspendapplet_1.5.0-3_arm.ipk]
- qpose-bin - [qpose-bin_3.5-0.2-1_arm.ipk]
- qpose-data - [qpose-data_3.5-0.2-2_arm.ipk]
- sambacontroller - [sambacontroller_0.1-0_arm.ipk]
- shell - [shell_latest_arm.ipk]
- smbmount - [smbmount_0.1_arm.ipk]
- smbmounter - [smbmounter_0.1-2_arm.ipk]
- smbpasswd - [smbpasswd_0.1-1_arm.ipk]
- snes9x_sdl - [snes9x_sdl-1_arm.ipk]
- sun personal-profile (j2me) - [personal-profile-for-zaurus_arm.ipk]
- tasklist-applet - [tasklist-applet_1.0.5_arm.ipk]

- vpnc - [vpnc_0.3.2-1_arm.ipk]
- zbasilisk - [zbasiliskii_0.3_arm.ipk]
- zbochs - [zbochs.tar.gz]
- zemufe - [zemufe_0.1.1-3ex_arm.ipk]
- zgnuboy - [zgnuboy_1.0.3-3_arm.ipk]
- znester - [znester_7.1-1_arm.ipk]

Libraries and Drivers

- bvdd - [bvdd_0.4.0-1_arm.ipk]
- ipsec-module - [ipsec-module_2.4.20-1_arm.ipk]
- kern-mod-squashfs - [kern-mod-squashfs_c3000-2.1-2_arm.ipk]
- kmicrokdelibs - [kmicrokdelibs_2.1.2_arm.ipk]
- libffmpeg - [libffmpeg_0.4.6_20030304_arm.ipk]
- libgc - [libgc_6.3alpha4-1_arm.ipk]
- libglade - [libglade_2.0.1-1_armv5tel.ipk]
- libfloat - [libfloat_1.0_arm.ipk]
- libiconv - [libiconv_1.8-2_arm.ipk]
- libmad - [libmad_0.14.2b-1_arm.ipk]
- libncurses - [libncurses_5.0_arm.ipk]
- libogg - [libogg_1.1.2-1_arm.ipk]
- libopie1 - [libopie1_1.1.0-20031220_arm.ipk]
- libopie2 - [libopie2_1.8.2-20031220_arm.ipk]
- libcap0 - [libpcap0_0.7.2-20031220_arm.ipk] [libpcap0_0.7.2_arm.ipk]
- libperl - [libperl_5.6.1_arm.ipk]
- libpng3 - [libpng3_1.2.4-1_arm.ipk]
- libsdl-image - [libsdl-image_1.2.5cvs-1_arm.ipk]
- libsdl-mixer - [libsdl-mixer_1.2.5cvs-1_arm.ipk]
- libsdl-net - [libsdl-net_1.2.5cvs-1_arm.ipk]
- libsdl 1.2.5 - [libsdl_1.2.5-slzaurus20050410_arm.ipk]
- libvorbis - [libvorbis_1.1.0-1_arm.ipk]
- libvorbisdec - [libvorbisdec_1.2.0-1_arm.ipk]
- libxml - [libxml2_2.6.14-1_arm.ipk]
- libyahoo2 - [libyahoo2_cvs-20040713_arm.ipk]
- qpe-libqtopia - [qpe-libqtopia_1.6.0-13_arm.ipk]
- openssl - [openssl_0.9.7d_arm.ipk]
- opie-mediaplayer-codecs - [opie-mediaplayer-codecs_0.2.2_arm.ipk]
- sandisk-plus - [sandisk-plus_0.1_arm.ipk]
- tun-module - [tun-module_2.4.20-1_arm.ipk]
- vga-presentation - [vga-presentation_1.0.1_arm.ipk]
- wlan-prism3 - [wlan-prism3_1.0.0_arm.ipk]
- zlib - [zlib_1.2.2-1_arm.ipk]

Games

- supertux - [supertux_0.1.2-3_arm.ipk]
- aliens - [aliens_1.0.0_arm.ipk]
- barrage - [barrage_1.0.2_arm.ipk]
- lgeneral - [lgeneral_1.2beta-2_arm.ipk]
- lmarbles - [lmarbles_1.0.7-2_arm.ipk]
- ltris - [ltris_1.0.10-2_arm.ipk]
- freeciv - [freeciv_zaurus_0_0_5_bin.tar.bz2]
- qfish2 - [qfish2_1.1.0-sl700_arm.ipk]
- tetrix - [opie-tetrix_1.5-1_arm.ipk]
- billiardz - [billiardz_0.1.2_arm.ipk]
- backgammon - [backgammon_0.6.1_arm.ipk]
- Jahtzee - [jahtzee_0.9.1_all.ipk]
- gomoku_moro - [gomoku_moro_1.0.2_arm.ipk]
- qpe-patience - [qpe-patience_1.5_arm.ipk]
- cards - [cards_1.0.0_arm.ipk]
- shisensho - [shisensho_1.0.0_arm.ipk]
- ksokoban - [ksokoban_1.5.0-16_arm.ipk]

- mahjongg - [mahjongg_1.0.0_arm.ipk]
- spades - [spades_1.0_arm.ipk]
- zsolitaire - [zsolitaire_1.02_arm.ipk]
- zknights - [zknights_0.6.1-1_arm.ipk]
- Eliminator - [Eliminator_1.0_arm.ipk]
- javachess - [javachess_1.0_arm.ipk]
- javanoid - [javanoid_1.54-2_arm.ipk]
- laserchess - [laserchess_1.09_arm.ipk]
- coffeeChess - [coffeeChess_0.0.1_arm.ipk]
- javello - [Javello_0.0.1_arm.ipk]
- zudoku - [zudoku-figlabs_1.1_arm.ipk]
- warp - [warp_1.0_arm.ipk]
- warpfleet - [warpfleet_0.1_arm.ipk]
- doom (doomdemo and prboom) - [doomdemo_1.8-1_arm.ipk] [prboom_2.2.3-2_arm.ipk]
- quake - [qpe-quake_1.5.0-2_arm.ipk] [qpe-quake-data_1.5.0-1_arm.ipk]
- heretic - [heretic-demo_1.2-3_arm.ipk] [heretic-engine_1.0.4-2_arm.ipk]
- ctux (cyclone tux) - [ctux_1.0.0_arm.ipk]
- dtux (dancing tux) - [dtux_1.0.0_arm.ipk]

Well, Tux isn't really a game but it is funny. You will need to install [qpe-libqtopia_1.6.0-13_arm.ipk] for tux.

A lot of applications written for prior Z versions still work for the C3000 and C3100, however, the screen is rotated to portrait by default for those applications which can be changed easily. Most require no changes to work, but some need some tweaking in order to work properly on the C3000 and C3100. Look in the customisation section for details for those.

Generally, feeds for Cacko and pdaXrom are also good sources for finding applications written for the older Sharp models. They will run on Qtopia and X/Qt respectively.

The Sharp ROM on the C3000 is not really a ROM image. It really is a misnomer carried on from previous models. The SL-C3000 doesn't really have a flashable ROM image with the full OS and applications on it. Those are stored on the MicroDrive (harddisk) instead on the C3000. It really is a Zaurus Linux distro packaged by Sharp. There are several other distros other than the Sharp one available for the Zaurus in various stages of development. I have created a dedicated section on alternate distros/ROMs for the SL-C3000 and SL-C3100. The customisaton section that follows is primarily for the stock Sharp distro but may also be applicable to Cacko since Cacko is an improved version of Sharp distro with various customisations and enhancements already applied. I have also created sub pages dedicated to customising pdaXrom and customising OpenZaurus for the SL-C3000 and SL-C3100.

In addition, you can also run Debian packages if you install Pocket Workstation. OpenOffice for example works under Pocket Workstation. See the X/Qt section for more details.

There are also lots of emulators available for the Zaurus with which you can run applications and games for Nintendo, GameBoy, AppleII, Palm and even DOS. You can even run Java applications and games from your mobile phone on the Zaurus with a MIDP enabled J2ME implementation.

I also build a few ipk packages to make it easier to customise the Zaurus. Here is a quick summary of them and what they do. The customisation section has further details. The packages are zipped (not for compression but to prevent them from getting corrupted) so you will have to unzip them before you can install them.

- [c3000-custom-jaen_0.2_arm.ipk] - add English to the menus
- [zicons-wmtux_0.2_arm] - icon package for replacing system icons and extra app icons
- [keyhelper-c3000map_0.4_arm.ipk] - enhanced keyboard mapping for C3000 and C3100
- [usbkbd-en_2.4.20_arm.ipk] - switch USB keyboard map when keyboard is plugged in or unplugged
- [dualkbd_2.4.20_arm.ipk] - allow shift num pad on USB keyboard to function like shift numeric keys
- [langswitch_0.2_arm.ipk] - switch between English and Japanese menus
- [bgswitch_0.1_arm.ipk] - switch between several wallpapers
- [netswitch_0.4_arm.ipk] - control for USB network with drivers
- [netswitch-lite_0.4_arm.ipk] - control for USB network (no drivers)
- [usblan-rtl8150_2.4.20_arm.ipk] - USB network driver (rtl8150)
- [usblan-pegasus_2.4.20_arm.ipk] - USB network driver (pegasus)
- [irnet_2.4.20_arm.ipk] - irnet driver and config for IrDA networking
- [bluetooth-support_1.23-1_arm.ipk] - bluetooth drivers and tools
- [bluetooth-gui-lite_1.23-1_arm.ipk] - bluetooth tools without drivers
- [ntfs-zaurus_2.4.20_arm.ipk] - NTFS driver for Zaurus

- [zmouse_0.1_arm.ipk] - enable mouse for the Zaurus
- [unicodefonts-verdana_1.5.0-3_arm.ipk] - rotatable unicode font verdana
- [libstdc6_1.2.2_arm.ipk] - standard C library supplement for Sharp ROM
- [libstdc5-compat-sharp_0.5_arm.ipk] - pdaXrom C library supplement compatible with Sharp ROM
- [xqt-debian-scripts_0.6.1_arm.ipk] - custom X start script
- [xqt-libXrender_1.2.2_arm.ipk] - updated X-render library
- [apachegui_0.1_arm.ipk] - apache control GUI
- [zflite-gui_0.2_arm.ipk] - flite GUI
- [filelaunch-en_0.4.5_arm.ipk] - (not fully translated yet, but usable)
- [opera-en-helper_0.0.2b_arm.ipk] - opera config GUI (translated from Japanese)
- [automounter-c3000_0.5.0_arm.ipk] - auto mounter for USB drives and loop filesystems
- [sdmmc-module_2.4.20_arm.ipk] - updated SD driver to support larger SD cards
- [usb-modules_2.4.20_arm.ipk] - additional USB device drivers
- [iptables-base_2.4.20_arm.ipk] - minimum iptables
- [iptables-additional_2.4.20_arm.ipk] - additional iptables
- [shorewall-c3000_1.4.5-1_arm.ipk] - packet filter firewall
- [kismet-misc_0.3_arm.ipk] - kismet script for GUI launch and sound files
- [j2me-zaurus_1.1.8_arm.ipk] - personal profile for zaurus j2me with swing and compiler
- [jlauncher_0.1_arm.ipk] - java launch wrapper for j2me
- [midp-launcher_0.2_arm.ipk] - enables MIDP games to run with Jeode
- [babbletower_0.9.3_arm.ipk] - babbletower dictionary reader (requires j2me)
- [dosbox_0.6.3-3_arm.ipk] - dos emulator
- [fbvncserver-c3000_0.9.4-0.2_arm.ipk] - hacked fbvncserver for C3000/C3100
- [qtopia-sysinfo_1.23-3_arm.ipk] - enhanced sysinfo tool with process and mount controls as well as more detailed disk info
- [qtopia-addressbook_1.23_arm.ipk] - addressbook with alphanumeric sorting support
- [qtopia-combbatteryapplet_1.0.6_arm.ipk] - updated battery applet with overclocking/underclocking support
- [qtopia-memoryapplet_1.0.4_arm.ipk] - updated memory applet with better swapfile management
- [qtopia-keyboardapplet_1.0.0_arm.ipk] - keyboard layout mapper applet
- [qtopia-network-usblan_1.0.0_arm.ipk] - network config for usb lan adaptor
- [qtopia-network-bluetooth_1.0.0_arm.ipk] - network config for bluetooth adaptor
- [qtopia-usbapplet_1.0.3_arm.ipk] - USB device control applet
- [japanese-support-c3100jaen_1.23_arm.zip] - add Japanese support to Cacko 1.23
- [pico_4.4_arm.ipk] - console based file editor
- [cpio_2.6_arm.ipk] - archiving tool
- [top_0.3.6_arm.ipk] - system tool
- [wget_1.10.2_arm.ipk] - download tool
- [qpPhoto_1.0.3_arm.ipk] - graphics tool modified for C3x00
- [yasump_0.40_arm.ipk] - MOD player modified for C3x00
- [ipktools_0.3.5_arm.ipk] - various scripts for manipulating ipk files including conversion of debian packages into ipk files
- [zgcc2-95-2-lite.zip] - minimal Zaurus on-board gcc development environment
- [zgcc2-95-2.zip] - Zaurus on-board gcc development environment
- [zgcc 2.95.3 (cramfs)]/[zgcc2-95-3 (squashfs)] - extended and updated Zaurus on-board gcc development environment for X/Qt

I have also made a section dedicated to X/Qt and Debian PocketWorkstation that is more generic and not just for the SL-C3000 and SL-C3100 models. It contains the following:

X/Qt Jumbo and Applications packages

- [xqt-gtk-jumbo_4.3-0.7.1_arm.ipk] - X/Qt jumbo package
- [firefox_0.9-3_arm.ipk] - Mozilla Firefox
- [thunderbird_0.6-3_arm.ipk] - Mozilla Thunderbird
- [xqt-gimp_1.2.5-3_arm.ipk] - The Gimp
- [abiword_2.0.0-2_arm.ipk] - AbiWord

X/Qt Jumbo and Application compressed images

- [xqt-gtk-jumbo.cramfs] - X/Qt jumbo image
- [xqt-apps.cramfs] - X/Qt applications image
- [xqt-mozilla.cramfs] - Firefox and Thunderbird image
- [xqt-openoffice.cramfs] - OpenOffice image
- [java.cramfs] - Blackdown JRE 1.3.1 image
- [xqt-install.sh] - the installer for the X/Qt images

Debian PocketWorkstation and OpenOffice

- [xqt-debian-jumbo-lite_4.3-0.7.1_arm.ipk] - X/Qt jumbo lite for Debian
- [xqt-debian-install.sh] - X/Qt Debian install script
- [zaurus-debian-jumbo-v18-b01.tar.gz] - Debian tarball
- [zaurus-debian-doc-v18.tar.gz] - Debian docs tarball
- [zaurus-debian-openoffice114.tar.gz] - OpenOffice tarball

In addition, I have updated and enhanced the X/Qt packages with additional and newer libraries so that they can run the latest pdaXrom applications. A separate section for these new updated packages can be found in a dedicated X/Qt super jumbo - pdaXQtrom section. This section is also generic and not just for the SL-C3000 and SL-C3100 models. With these packages installed, you can run the latest X based applications under Sharp/Cacko.

pdaXQtrom Application packages

- [abiword_2.4.0-2_arm.ipk] - Word Processor
- [aspell_0.60.4_arm.ipk] - Spell Checker
- [axyftp_0.5.1_armv5tel.ipk] - FTP Client
- [bluefish_0.13_armv5tel.ipk] - Text Editor
- [denemo_0.5.3_arm.ipk] - Sheet Editor
- [dia_0.92_armv5tel.ipk] - Diagrams
- [dillo-xft_0.8.5_arm.ipk] - Web Browser
- [dosbox-x11_0.6.3-3_arm.ipk] - DOS Emulator
- [epdfview_0.1.2_arm.ipk] - PDF Viewer
- [ethereal_0.10.9-1_armv5tel.ipk] - Packet Analyzer
- [firefox_1.5_arm.ipk] - Web Browser
- [fltdj-utf8_0.7_armv5tel.ipk] - PIM
- [free42_0.3_arm.ipk] - HP Calculator
- [gaim_2.0-b3_arm.ipk] - IM Client
- [galculator_1.2.5_arm.ipk] - Calculator
- [gftp_2.0.18_arm.ipk] - FTP Client
- [gimp_2.3.4_arm.ipk] - Graphics Tool
- [gnumeric_1.6.0_arm.ipk] - Spreadsheet
- [gpaint2_0.2.3_arm.ipk] - Image Editor
- [gpe-edit_0.13_arm.ipk] - Text Editor
- [gpe-filemanager_0.23_arm.ipk] - File Manager
- [gpe-gallery_0.97_arm.ipk] - Image Viewer
- [gpe-soundbite_1.0.6_arm.ipk] - Image Viewer
- [gpe-word_0.2_arm.ipk] - Word Processor
- [gplflash_0.4.13_arm.ipk] - Flash Player
- [gps_1.1_arm.ipk] - Graphical ProcessView
- [gpsdrive_2.10pre3_arm.ipk] - Global Positioning
- [gqview_2.1.1_arm.ipk] - Image Viewer
- [grisbi_0.5.9_arm.ipk] - Personal Finance
- [gtkyahoo_0.18.2_arm.ipk] - Yahoo IM Client
- [gyach_0.9.4_arm.ipk] - Yahoo IM Client
- [hp48_0.2_arm.ipk] - HP48 Emulator
- [knowde1.8.0_arm.ipk] - Knowledge Management
- [leafpad_0.7.9_arm.ipk] - Note Pad
- [links_2.1_arm.ipk] - Web Browser
- [mc_4.6.1-pre1_arm.ipk] - File Manager
- [minimo_1.7.12_armv5tel.ipk] - MiniMozilla
- [mplayer-1.1pre8_arm.ipk] - MPlayer for X
- [multi-aterm_0.2.1] - Terminal Emulator
- [nedit_5.4_armv5tel.ipk] - Text Editor
- [planner_0.13-1_armv5tel.ipk] - Project Management
- [putty_0.58_arm.ipk] - SSH Client
- [scite_1.62_arm.ipk] - Text Editor
- [smessy_0.1.1_arm.ipk] - SMS Messenger
- [stardict_2.4.3_arm.ipk] - Dictionary
- [sylpheed-gtk2_2.0.2_arm.ipk] - Email Client
- [thunderbird_1.0.7_arm.ipk] - Email Client
- [tightvnc_1.2.9-1_arm.ipk] - VNC Client
- [xarchiver_0.3.1_arm.ipk] - Archive Tool
- [xchat_2.6.0_arm.ipk] - IRC Client

- [xmms_1.2.10-4_arm.ipk] - Media Player
- [xpad_2.11_arm.ipk] - Sticky Pad
- [xpdf_3.01-4_arm.ipk] - PDF Viewer
- [xpdf-tools_3.01-4_arm.ipk] - PDF Extraction Tools
- [xournal_0.3.1_arm.ipk] - Journal App

pdaXQtrom Games packages

- [cgoban_1.9.12_arm.ipk] - gnu go
- [gnuchess_5.07_arm.ipk] - chess
- [gsoko_0.4.2_arm.ipk] - sokoban
- [gtkatlantic_0.4.0_arm.ipk] - monopoly client
- [xbomb_2.1a_arm.ipk] - minesweeper
- [xdemineur_2.1.1_arm.ipk] - minesweeper
- [xkobo_1.11+w01_arm.ipk] - space game
- [xpuyopuyo_0.9.8_arm.ipk] - puzzle
- [xshogi_1.3_arm.ipk] - gnu shogi
- [xsokoban_3.3c_arm.ipk] - sokoban

Getting Started:

Before embarking on customisation and hacking your Zaurus, you need to know how to handle it. There is a very good manual written by TRIsoft on this topic. Please refer to TRIsoft's C3000 manual which they provide for free:

- http://trisoft.de/pdf/c3000qs.pdf - the original TRIsoft quickstart manual in German
- http://trisoft.de/pdf/c3000qse.pdf - the translated TRIsoft quickstart manual in English

The TRIsoft quickstart manual is a very good summary on all the important things you need to know in order to operate your Zaurus. I personally prefer their original German version. Their English translation is not perfect but it is much easier to read than the original Sharp manual which is in Japanese. The TRIsoft manual concisely summarises just the important facts that you need.

User Manual:

Once you have learned how to handle your Zaurus and discovered some of its features, you probably want to learn how to use the applications that come with the Zaurus. This shouldn't be a problem for most savy users, but nevertheless, FigLabs have taken the time and written a complete user manual for the C3000, and they have made it available online: http://www.figlabs.com/catalog/ug.php

---

Customisations:

Most information about the SL-C3000 found on the net is in Japanese and a lot of the instructions out there are for earlier models of the Zaurus. Although a lot of that info is still valid because the C3000 is backward compatible with lots of the older models not everything works. Since the C3000 is newer, it has extra capabilities those guides do not mention. Also sometimes things that worked in the older version

do not work on the C3000 anymore due to it being implemented differently on the C3000. This guide is intended specifically for the SL-C3000 model (also sometimes referred to as Spitz) but a lot of the info can also be applied to similar models. Most of the information will also apply to the SL-C3100 (also known as Borzoi) since the two models are very similar. In some areas, however, there are significant differences between these two models and those will be highlighted.

Some of the customisations require additional files which should be downloaded first. Create a folder called *custom* on a CF/SD card or via the USB connection to /home/zaurus/Documents and copy these customisation files into that folder. Most the instructions below assume that the files have been transferred to */home/zaurus/Documents/custom*

The first thing you probably need to do is to install a terminal/console application and **backup your Zaurus**. These instructions are primarily for the Sharp distro/ROM, although most of it can be used for Cacko as well.

Essential Packages

The following are applications and utilities that I consider absolutely essential and must have packages:

- qkonsole - a terminal console with multiple tabs (sessions), scrollbars, colour selection, history and fonts
- keyhelper - allows you to customise and reassign keys
- c3000-custom-jaen - add English text to the menus and tabs
- opie-sh and/or qshdlg - shell enhancement for dialog and input boxes (required by some apps)
- sudo - enhances security if used correctly (needed by some apps)
- unicodefonts-verdana - extra font with extended character set and rotatable
- automounter-c3000 - enables automatic mounting of loop and USB devices
- Tetsu's special kernel - an enhanced kernel for Sharp ROM/distro that fixes some annoying bugs and improves performance as well

The following are very useful applications that I would also always install:

applets that appear in the taskbar

- clipboard-applet - allows you to cut and paste between application
- qpe-suspendapplet - allows you to enable and disable suspend temporarily
- tasklistapplet - allows you to select running applications from a list
- memoryapplet - allows you to manage swap files and view memory usage
- combbattery-applet - allows you to manage and monitor power settings as well as overclock and underclock

These libraries are often needed by several applications and should be installed to satisfy most generic dependencies:

libraries

- kmicrokdelibs - [kmicrokdelibs_2.1.2_arm.ipk]
- libfloat - [libfloat_1.0_arm.ipk]
- libiconv - [libiconv_1.8-2_arm.ipk]
- libncurses - [libncurses_5.0_arm.ipk]
- libpng3 - [libpng3_1.2.4-1_arm.ipk]
- libsdl - [libsdl_1.2.5-slzaurus20050731_arm.ipk]
- qpe-libqtopia - [qpe-libqtopia_1.6.0-13_arm.ipk]
- openssl - [openssl_0.9.7d_arm.ipk]
- zlib - [zlib_1.2.2-1_arm.ipk]

These are very useful runtimes and often, other useful services or applications depend on having these runtimes:

runtimes

- samba - allows sharing of files over network
- jeode - java runtime
- perl - perl interpreter

- bvdd - enhanced video driver
- mplayer - media player supporting many video formats
- xqt-gtk-jumbo - X windows system for Qtopia
- debian-pocketworkstation - embedded fully functional Debian environment

These command line tools are very important. They provide funtionality that one would expect from any OS nowadays.

command line tools

- vim or pico - console text editors
- file - tells you the file type according to mime settings
- dos2unix - fixes return characters between DOS and UNIX systems
- unzip - allows you to unzip files
- zip - allows you to create zip files
- bzip2 - uses the newer and highly more compressable bzip2 format
- wget - command line http client
- openssh-client - secure terminal client

applications

- qpdf2-freetype - pdf reader
- yedit - text editor
- visualq - graphics editor
- qazoo - yahoo messenger clone
- zicic - irc client
- kino2 - mplayer frontend
- firefox - mozilla browser
- thunderbird - mozilla email client
- openoffice - open source office application

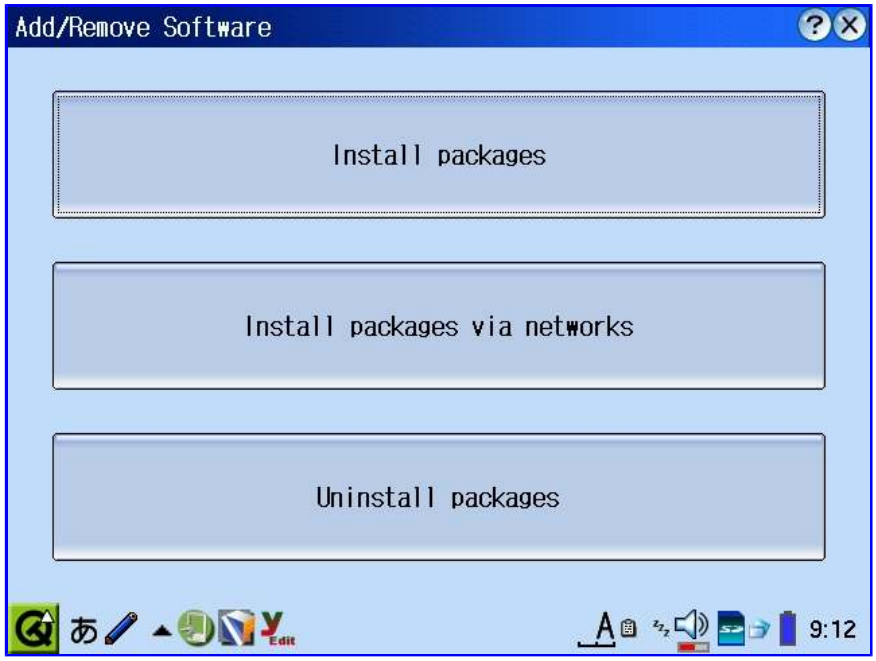
Some tweaking of the look and feel is also required. The default Qtopia theme shipped by Sharp is not very pretty (depends on taste). Changing the theme to something like Crystal-Blue will make it look much better. There are several themes to choose from out of the box, but many more themes can be downloaded and installed.

In addition, the default icons and background can be changed also. Install some icon packages and get some nice backgrounds. **zicons-wmtux** contains my favourite icons. There are also plenty of nice backgrounds. **Plasterer** can be used to ensure that the background looks consistent even when the screen is rotated. Finally, replace the ugly default screensaver with **LUSScreensaver**.

More details on how to do all this is described in details later.

Installing Packages

Packages can be installed and uninstalled via the Package Installer tool (qinstall) under the Settings tab. By default, the Zaurus installs applications to main memory which is the /hdd2 partition on a C3000 and the /home partition on a C3100. You can also install applications to your SD or CF card as well if the application allows it. Some applications can only be installed to main memory, wheras others allow you to either install to SD or CF as well if they are formatted as ext2 or ext3. There are also applications that can be installed to SD or CF cards that have a FAT filesystem.



In addition, with a console, you can use the **ipkg** command to install and uninstall applications as well, eg:

```
# ipkg install ipkfile
# ipkg remove ipkfile
```

The advantage of the ipkg tool that it can be used to script a batch of applications, such as all your favourite fonts, or all your security tools. It also shows you detailed error messages if something goes wrong unlike the qinstall tool which just reports an error but nothing else useful.

However, the ipkg tool does not generate the required links for you if you want to install to SD or CF card. Qinstall (the GUI Package Installer) does that for you during the installation if you select either SD or CF for the destination location. You will need to run ipkg-link after installing to SD or Cf card in order to relink the applications you install with ipkg. However, ipkg-link does not get shipped with the C3000 nor the C3100 by default. I have included ipkg-link in my ipktools package.

By default, the package installer expects package files to be located under /hdd3/Documents/Install_Files, /mnt/card/Documents/Install_Files and /mnt/cf/Documents/Install_Files. ipkg exspects all the packages to be in the same directory. Additional sources and package feeds can be configured by modifying /etc/ipkg.conf and adding source location like the following example:

```
src zug http://www.zaurususergroup.org/feed/
```

I have created a script called **xipk** (part of my ipktools package) which allows you to install packages to virtually any location you like. This allows you to install packages to /hdd3 where the bulk of the MicroDrive's space is instead of the default main memory which is /hdd2 on a C3000 and /home on a C3100. However, not all applications can be installed to /hdd3 because it is by default formatted as a FAT filesystem and can't handle symbolic links. If you reformat it as ext2 or ext3, then you won't have that problem (see later sections on /home and loopback filesystems).

Some application that you install will be in Japanese. Have a look at the Localisation section to see if you can change it to English. Try the **langswitch** tool, which can fix it for you in most cases.

Also, if the application was written for an older Zauri model, then you need to change the application's default screen orientation. See the Screen Orientation section on how to do this.

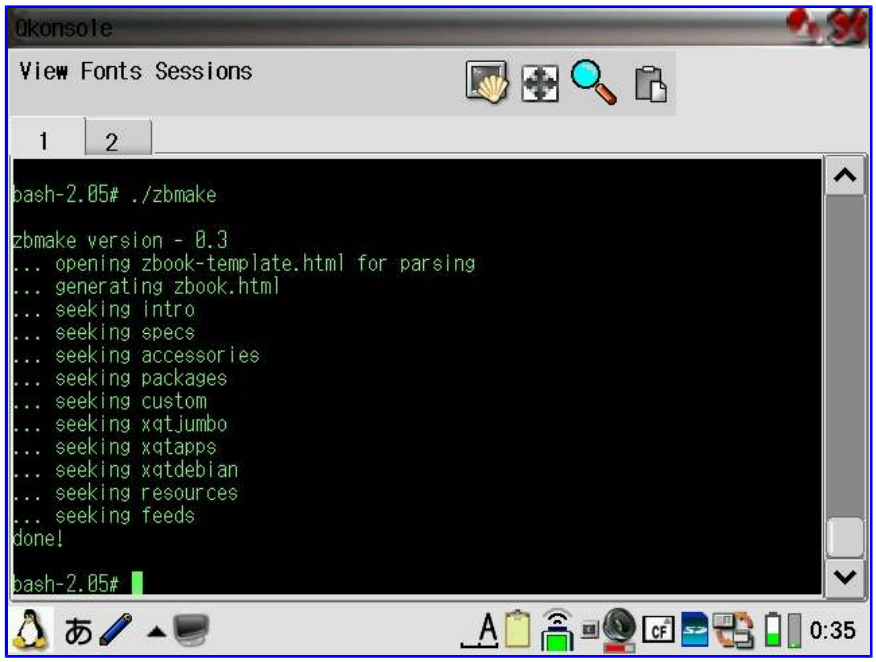
Installing a terminal

Connect the Zaurus via USB (make sure to plug the USB cable into the PC first, then the Zaurus). Copy

the terminal ipk file [qpe-terminal-ja_1.5.0-3_arm.ipk] from the CD-ROM into the /Documents/Install_Files directory. Disconnect the USB device on the PC. The Zaurus will turn back to normal mode. Click on the third TAB on the top and then click on the little disk like icon. Then select the Install_Files folder. Inside, click on the ipk file (should be the only file there) and the installer will launch it. Press install (the big button at the bottom) and then OK (the button on the left) on the dialog box that comes up.

If you have a SD or CF card and a card reader for it, then you can also just copy the ipk file onto the SD card and insert the card into the Zaurus afterwards and install the ipk file from the card instead of using the USB cable which sometimes can be problematic.

A much better terminal [qkonsole_0.9.3-20040205_arm.ipk] can be download and installed instead of the one provided on the CD-ROM.



Most of the customisation work requires a terminal so you really should install one. All the instructions inside a white box assume its done from within a terminal window, and most entries in gray boxes are illustrations of configuration file fragments. Also I assume you know how to use **vi**. Personally, I love Vim (vi improved) and use it all the time for most things. However, if you are really struggling to use **vi**, then you can use **pico** instead which is like the DOS edit (see pico section on how to install). Then whenever you see the instructions tell you to use vi to edit or create a file, use pico instead. The *esc* key in vi is mapped to the *cancel* key on the Zaurus.

Localising/Converting to English

The C3000 and C3100 come in Japanese only by default. The irony of this is that Qtopia and most of Linux were developed in English, and Sharp had to change it all to Japanese, and we have to change it all back again. This makes changing the Zaurus back into English rather simple except for a few new applications that were written in Japanese natively. There are several approaches to switch back into English only mode, and there are even some scripts out there that automate the whole process. I consider the Japanese a bonus so no way am I going to get rid of it. (Ever tried to add Japanese support to an older Windows version?)

Switching back to English (quick and dirty):

Launch the terminal and change the /home/zaurus/Settings/locale.conf file to use 'en' instead of 'ja'.

```
[Language]
Language = en
[Location]
Timezone = Australia/Sydney
```

Reboot the Zaurus.

Localising to English but keeping Japanese:

(English menus, mixed Japanese and English titles)

The goal of this localisation is to keep all the Japanese funtionality but have English menu items and mixed Japanese and English display of tab entries. Japanese input method, fonts and dictionary will not be affected by this customisation and will still work afterwards.

```
# su
# cd /home/QtPalmtop/i18n/ja
# mkdir .hide
# mv *.qm* .hide
# mv .hide/libjpn* .
# cp /home/zaurus/Documents/custom/movieplayer.qmid .
# chown root:qpe movieplayer.qmid
# chmod 640 movieplayer.qmid
# cd /home/QtPalmtop/i18n/en
# cp /home/zaurus/Documents/custom/movieplayer.qmid .
# cp /home/zaurus/Documents/custom/libsl.qmid .
# chown root:qpe *.qmid
# chmod 640 *.qmid
# cd /home/QtPalmtop/bin
# mv word-eucJP.rc word-eucJP.rc.hide
# cd /home/QtPalmtop
# tar cf apps-orig.tar apps
# gzip apps-orig.tar
# cp /home/zaurus/Documents/custom/apps-mod.tar .
# tar xf apps-mod.tar
# chown -R root:qpe apps
# cd /home/QtPalmtop
# tar cf etc-orig.tar etc
# gzip etc-orig.tar
# cp /home/zaurus/Documents/custom/etc-mod.tar .
# tar xf etc-mod.tar
# chown -R root:qpe apps
# cd /home/zaurus
# tar cf Settings-orig.tar Settings
# gzip Settings-orig.tar
# cp /home/Zaurus/Documents/custom/Settings-mod.tar .
# tar xf Settings-mod.tar
# cd Settings
# chown zaurus:qpe *.conf
# chmod 644 *.conf
```

Reboot the Zaurus and this is what it will look like:

This package [c3000-custom-jaen_0.2_arm.ipk] will do the above and is what I use to recover my localisation if I need to reset my Zaurus to factory setting and start again. This package works for the C3100 and C1000 as well. You can switch back to Japanese by simply uninstalling the package.

Some applications that you install after the localisation will still appear in Japanese, but most of those can be easily localised as well by looking under the /home/QtPalmtop/i18n/ja directory and if there is an additional qm file named after the application you installed there, simply move it into the .hide directory. However, some applications are written natively in Japanese and cannot be easily localised. I have created [langswitch_0.2_arm.ipk] which will allow you to move the qm files back and fro from the GUI (it requires opie-sh).

The help files are still in Japanese. If that bothers you and you want English help instead, then you can install helpfiles_1.23-lite-1_arm.ipk from the Cacko feed and once installed, create a link from ja to en under the help directory:

```
# su
# cd /home/QtPalmtop/help
# mv ja ja.orig
# ln -s en ja
```

The addressbook will display text and menus in English, but entries are still sorted according to the Japanese kana. If you rather have them sorted alphabetically and don't care about storing entries in Japanese, then you can install [qtopia-addressbook_1.23_arm.ipk].

If you want Netfront to be able to display German umlauts, French accents, and other special characters in addition to the standard English characters and Japanese characters, change /home/zaurus/Applications/netfront3/prefs and find an entry *FontFamilyJa:*. Add or modify *FontFamilyEn:* and set it to a font such as *verdana* that contains the extended character sets. The following font package is recommended: unicodefonts-verdana_1.5.0-3_arm.ipk

```
FontFamilyEn: verdana
FontFamilyJa: lcfont
```

Fonts

The Zaurus comes with the following fonts already pre-installed:

- lcfont
- fixed
- helvetica (same as mico-unicodefonts-helvetica_1.5.0-1_arm.ipk)

- micro
- smallsmooth
- smoothtimes

Unfortunately, the Japanese character mapping has some overlaps and it is not always possible to correctly map some extended characters in the latin character maps. Unicode fonts help a bit in this aspect. However, many fonts do not have all the unicode characters which results in little square boxes being displayed. It is essential to have a font which has all the unicode characters fully and correctly mapped, however, such a font will use over 1MB of memory for each font size.

The following fonts get very close to that, however, the unifont only has size 16.

- unismall_1.0.0_arm.ipk
- unifont_1.0-1_arm.ipk

Having fonts that contain as many character sets as possible is a good start, however, it also depends on the application whether it uses unicode and can extract the right character out of the fonts and display them. Some fonts are also missing details for screen rotation and thus will look garbled when the screen is rotated. Make sure you install the rotated font also if you find the font garbled on rotation.

I have added the following extra fonts:

- unicodefonts-verdana_1.5.0-3_arm.ipk
- mico-unicodefonts-georgia_1.5.0-2_arm.ipk
- mico-unicodefonts-utopia_1.5.0-1_arm.ipk
- vga-console-font_1.0-1_arm.ipk
- fonts-bitstream-vera-sans-mono-50_1.1_arm.ipk
- fonts-bitstream-vera-sans-mono-75_1.1_arm.ipk
- FreeSerifFont_20031008_all.ipk
- FreeSansFont_20031008_all.ipk
- FreeMonoFont_20031008_all.ipk
- misaki8_0.04_arm.ipk
- naga10_0.02_arm.ipk
- shnm12_0.03_arm.ipk
- shnm14_0.03_arm.ipk
- shnm16_0.05_arm.ipk
- ayu18_0.02_arm.ipk

Note: The unicodefonts-verdana_1.5.0-3_arm.ipk is a repackaged version of mico-unicodefonts-verdana_1.5.0-2_arm.ipk with a rotatable font set.

These fonts are for the default Qtopia deskop and applications. X/Qt and Pocketworkstation use a different set of fonts. The Qtopia fonts are stored in /opt/QtPalmtop/lib/fonts. Zaurus fonts use the Trolltech's QT Prerendered Format (QPF). If you want to make your own additional fonts, then you can convert fonts to Zaurus fonts by using a utility called makeqpf-arm which is provided by Trolltech.

lcfont is the default system font used with the Japanese system, ie when your locale is set to *ja* which is what your Zaurus is set to by default. However, Qtopia defaults to helvetica if it can't find the font it needs.

When generating fonts, you will need to generate two versions, one for portrait mode and one for landscape. The qpf font for the rotated screen has *t10* appended to the filename.

Here are some sample steps to convert the arial.ttf font to a size 16 qpf in landscape and portrait mode. Doing this may result your Zaurus being locked up, and you definately have to reboot your Zaurus afterwards so make sure you save any open files before doing the following:

```
# mkdir -p /hdd3/build/lib/fonts
# cp arial.ttf /hdd3/build/lib/fonts
# export QTDIR=/hdd3/build
# cd $QTDIR/lib/fonts
# echo "arial arial.ttf FT n 50 160 s" > fontdir
# makeqpf-arm -display Transformed:Rot0 -A
# makeqpf-arm -display Transformed:Rot270 -A
# cd /hdd3/build
# newipk arial
```

```
# cp /hdd3/build/lib/fonts/*.qpf /hdd3/arial/data/opt/QtPalmtop/lib/fonts
# makeipk arial
# su
# reboot
```

The above assumes you have downloaded makeqpf-arm from trolltech and extracted it to a location in the PATH, ie /usr/local/bin. Also it is assumed you have ipktools installed.

And finally, the Japanese display character for the / symbol is ¥. It is incorrectly displayed but the correct character is being used, so don't worry about it.

Key Mappings

The Zaurus comes with a full QWERTY keyboard and like most notebooks, some special characters and keys need to be accessed with a function key (Fn) combo. Most of those combos are already clearly marked on the keyboard, however, some are not marked and some are missing. All of the following key mappings work for most applications, however, some applications that have codepages directly compiled into them will not recognise the mappings and ignore them.

The Print Screen key sequence is:

- *Fn* + *Shift* + *c*

Navigation keys:

- *Fn* + *up arrow* = Page Up
- *Fn* + *down arrow* = Page Down
- *Fn* + *left arrow* = Home
- *Fn* + *right arrow* = End

Other useful and unmarked keys:

- *Shift* + - = `

There are other essential keys that need to be mapped. For that install keyhelper [keyhelper_1.2.2-1_arm.ipk] and put keyhelper.xml into /home/zaurus/Settings and then reload the key mappings from a console.

```
# cd /home/zaurus/Settings
# cp /home/zaurus/Documents/custom/keyhelper.xml .
# khctl reload
```

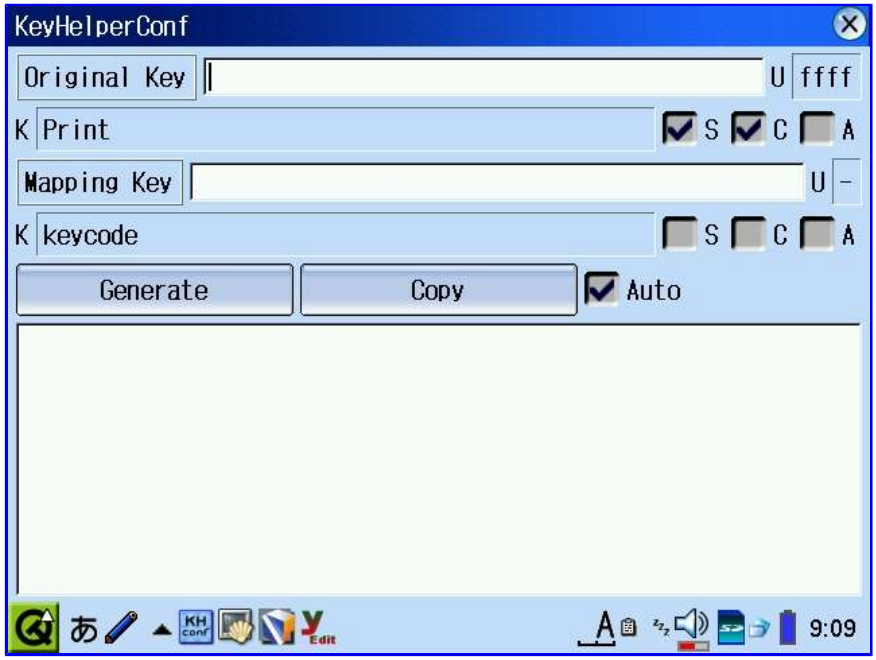
Installing keyhelper-c3000map [keyhelper-c3000map_0.4_arm.ipk] will also do the above and it will also enable the keyhelper menus too as shown below:

It will give you the following mappings:

- Sticky *Shift* key - press the *Shift* key and the next key you press will be shifted
- Sticky *Fn* key - press the *Fn* key and the next key you press will be the blue one on the top of each key
- Sticky *Ctrl* key - press the *Ctrl* key and the next key you press will be the combined with the Ctrl key
- *Alt* key - the left Japanese key (kana/hira) next to the Ctrl key
- *Shift+Mail* will bring up the application/quick menu for favourite apps
- *Shift+Address* will bring up the documents menu for frequently accessed files
- *Shift+Calendar* will bring up the settings menu for common tasks
- *Home* = switch between Menu Tabs
- *Shift+Home* will switch to the next application
- *Menu* key - brings up the pulldown menu of the current application, or launch the application from the quick menu when pressed with the shortcut key.
- *Shift+Menu* will bring up the task selector similar to alt + tab on windows
- *Ctrl* + *Menu* when *Menu* does not work, will give Alt + f (bring up the file menu)
- Swapped **/** and **,** key
- *Fn* + *o* = **{**
- *Fn* + *p* = **}**
- *Ctrl* + *t* = **«** (left double angle quotation)
- *Ctrl* + *y* = **»** (right double angle quotation)
- *Ctrl* + *w* = **×** (multiplication sign)
- *Ctrl* + *r* = **÷** (division sign)
- *Ctrl* + *-* = **±** (plus minus sign)
- *Ctrl* + *m* = **º** (little circle sign)
- *Ctrl* + *q* = **´**
- *Fn* + *q* = **`** (this one is redundant but I don't like the shift - combo)
- *Ctrl* + *a* = **ä** (umlaut a)
- *Ctrl* + *Shift* + *a* = **Ä** (umlaut A)
- *Ctrl* + *o* = **ö** (umlaut o)
- *Ctrl* + *Shift* + *o* = **Ö** (umlaut O)
- *Ctrl* + *u* = **ü** (umlaut u)
- *Ctrl* + *Shift* + *u* = **Ü** (umlaut U)
- *Ctrl* + *i* = **î** (circumflex i)
- *Ctrl* + *Shift* + *i* = **Î** (circumflex I)
- *Ctrl* + *e* = **æ** (ae)
- *Ctrl* + *Shift* + *e* = **Æ** (AE)
- *Ctrl* + *s* = **ß** (eszett)
- *Ctrl* + *Shift* + *d* = **Ð** (D with eth)
- *Ctrl* + *1* = **è** (e with grave)
- *Ctrl* + *2* = **È** (E with grave)
- *Ctrl* + *3* = **é** (e with acute)
- *Ctrl* + *4* = **É** (E with acute)

- *Ctrl + 5* = **ê** (e with circumflex)
- *Ctrl + 6* = **Ê** (E with circumflex)
- *Ctrl + 7* = **ë** (e with diaeresis)
- *Ctrl + 8* = **Ë** (E with diaerisis)
- *Ctrl + 9* = **ç** (c with cedilla)
- *Ctrl + 0* = **Ç** (C with cedilla)
- *Fn+Shift* will allow Alt key combinations from a-z and 0-9 except for x, c and v (the Alt key does not work for some applications)

The file /home/zaurus/Settings/keyhelper.conf contains the menu item definitions for what to display and what to execute. If you want to customize your own key mappings, you can install keyhelperconf which helps you determine the correct xml code required for the mappings:



You can also remap the application keys on the right hand side of the screen and at the bottom of the keyboard. The ApplicationKey tool under the Settings tab will allow you to assign different applications to those keys.

The following is my mapping:

- Dictionary -- Dictionary
- Calendar -- Qkonsole
- Address -- MusicPlayer

- Mail -- NetFront
- Home -- Home
- Menu -- Menu

And here are some useful keyboard shortcuts for Qkonsole:

- *Fn+s* will switch between terminals/consoles
- *Fn+n* will create a new terminal/console
- *Fn+5* will toggle fullscreen terminal/console
- *Shift+Up* will scroll up
- *Shift+Down* will scroll down

If you get ~0, ~1, ~2, ~3 when you hit the function keys at the bottom of your keyboard while in qkonsole, copy my modified linux.keytab and vt100.keytab to /opt/QtPalmtop/etc/keytabs and it won't happen anymore.

```
# su
# cd /opt/QtPalmtop/etc/keytabs
# cp /home/zaurus/Documents/custom/linux.keytab .
# cp /home/zaurus/Documents/custom/vt100.keytab .
```

Here are some useful keyboard shortcuts for NetFront:

- *Fn+h* home
- *Fn+r* refresh
- *Fn+s* save
- *Fn+u* view previous link
- *Fn+i* view next link
- *Fn+d* add bookmark
- *Fn+f* find
- *Fn+k* toggle search bar
- *Fn+m* new tab
- *Fn+b* close tab

The *esc* key in vi is mapped to the *cancel* key on the Zaurus.

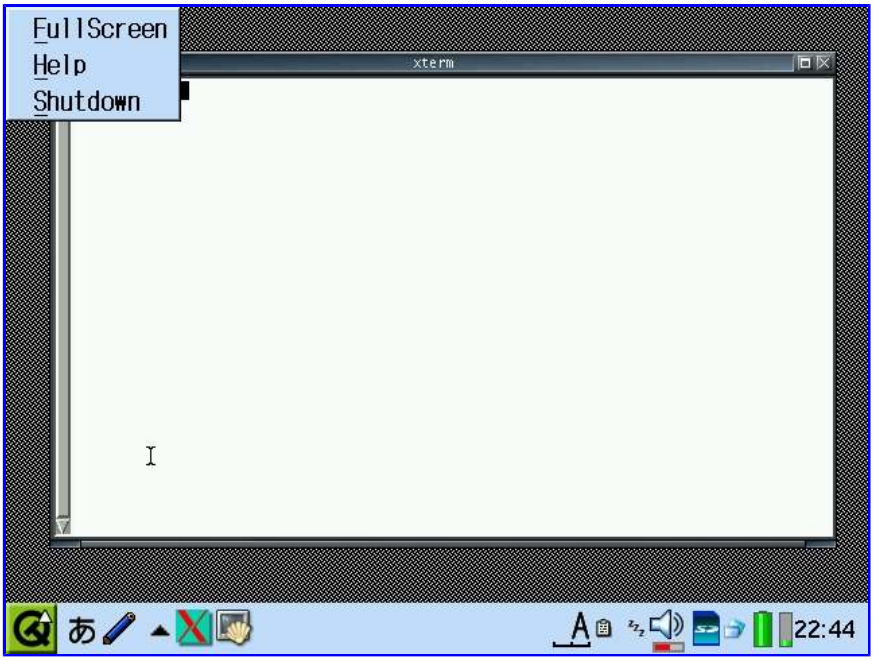
The function keys F1 - F10 in midnight commander are mapped to *Fn* + 1 to 10 respectively.

X/Qt (and also Debian PocketWorkstation which uses X/Qt) has its own keymapping. Get xmodmaprc-c3000 and place it under /home/zaurus and/or /home/root as .xmodmaprc if your keys misbehave.

.xmodmaprc does the following remapping so the keys are mapped similar to keyhelper:

- Swapped **/** and **,** key
- *Fn + q* = **`**
- *Ctrl + q* = **'**
- *Fn + o* = **{**
- *Fn + p* = **}**

The *Menu* key will activate the X/Qt control.



Fn + *m* will kill X/Qt

If your keys turn all CAPS then just hold down the Shift key for a few seconds and they will turn back to lower case.

File Associations

The file association information is derived from two places. The file */opt/QtPalmtop/etc/mime.types* stores the mime types that is used to derive the association information which is then combined with the desktop files stored under */opt/QtPalmtop/apps*

The information inside *mime.types* specifies what file extensions are associated with each file type. The desktop file for each application then specifies which file type the application should handle. If you have more than one application associated with a certain file type, then only one of them will be associated, however, the logic that the Z uses to determine which one to use is unknown to me. Therefore, make sure that only one of them is associated to a file type and you will get the expected file association.

As an example, to associate Opera to .htm and .html files, you would have the following in mime.types

```
text/html html htm
```

And opera.desktop would have the following:

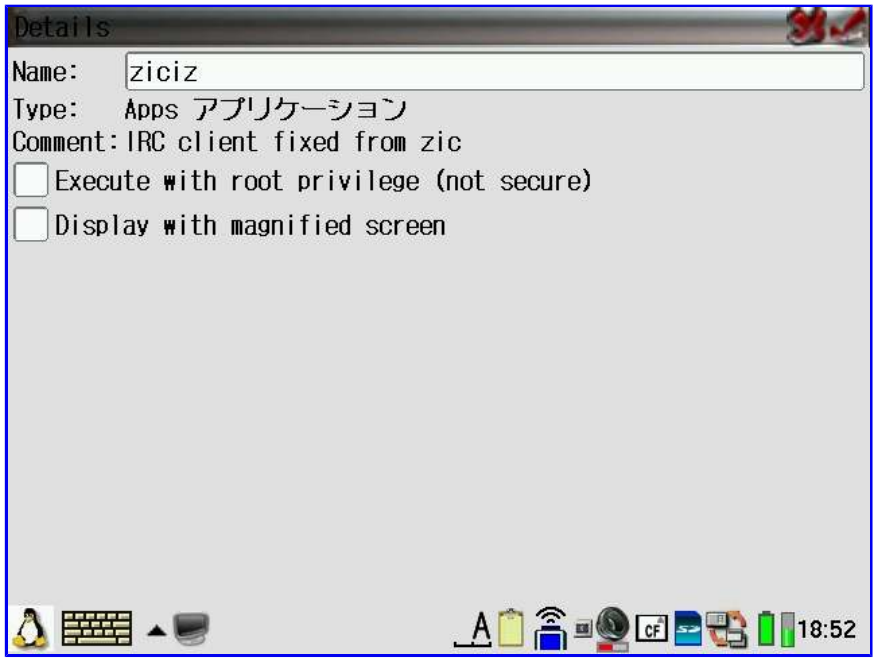
```
MimeType=text/html
MimeTypeIcons=opera
```

You can also associate an icon to the file association as well which is shown in the above sample. Multiple associates can be delimited with a semi colon (;).

Screen Orientation

The C3000 and C3100 have a larger screen resolution (640x480) than the older models. Older applications designed for those models by default start in portrait mode (because that is their default mode). The C3000 and C3100 support both portrait and landscape mode. The clam shell design detects when the screen is rotated and automatically re-adjusts the orientation. However, it also detects that these older applications were build for portrait mode and also automatically switches to portrait mode

when those applications are run.



To change this behaviour, tab on the application icon and hold the stylus there for a few seconds and a properties screen will appear. Untick the option "Display with magnified screen". (This needs to be done for each application)

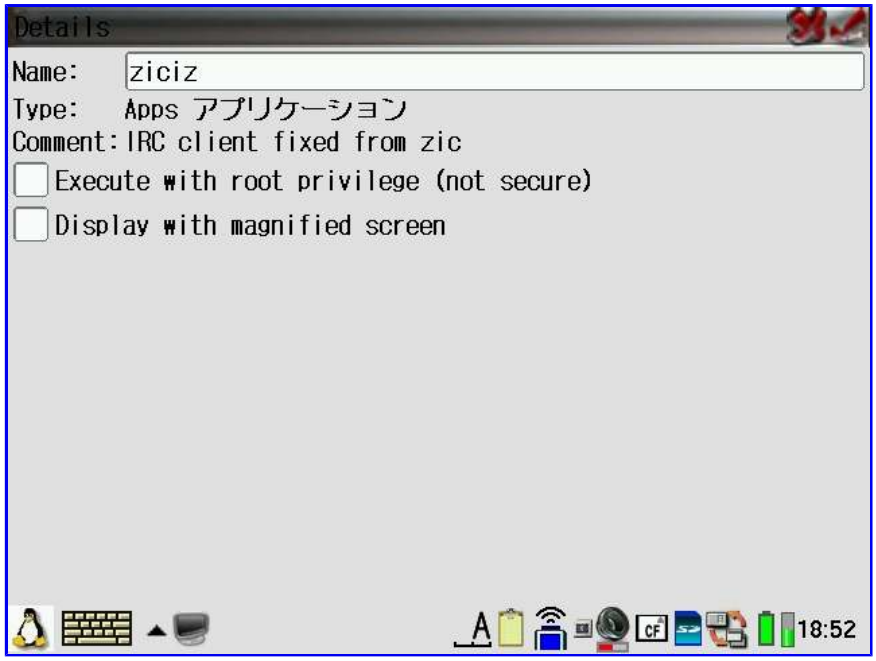
Alternatively you can add the following line to the appropriate .desktop file located under /opt/QtPalmtop/apps

```
Display=640x480/144dpi,480x640/144dpi
```

Only applications that do not have **Display=640x480/144dpi,480x640/144dpi** in their desktop file will show the *Display with magnified screen* option, however, if *EnableForcedVGA* in /home/zaurus/Settings/Launcher.conf is set to 0, then this option will not be available. The value of 3 in /home/zaurus/Settings/display.conf also means that the application will run in 640x480 mode.

Run as root

By default, applications run as the zaurus user, however, some applications need to be run with root priviledges.



To do that, tab on the application icon and hold the stylus there for a few seconds and a properties screen will appear. Tick the option "Execute with root priviledge".

Application Preloading

Some applications are preloaded on startup. This means when the Zaurus starts up, they are automatically loaded into memory. This makes them load very fast when you run them because they are already loaded. However, because of that, they also use up memory.

You can prevent them from preloading and conserve memory by disabling the preloading flag for each of the preloaded applications. To do that, tab on the application icon and hold the stylus there for a few seconds and a properties screen will appear. Untick the option "Fast load (consumes memory)".

Startup Screen Customisation

Customising the startup screen is possible. The startup image is a 90 degrees rotated bmp file (480x640) with a 24bit colour depth called Startup_screen.bmp. You simply need to copy it to the correct location so it will be picked up at boot time.

```
# su
# cp /home/zaurus/Documents/custom/Startup_screen.bmp /opt/QtPalmtop/pics144
```

It will look like this:

If you want to get rid of the Sharp screen right at the beginning of the boot process, then you will need to recompile the kernel with the sharp logo option disabled. Tetsu's kernel does that as well.

The file /root/etc/rc.d/rc.sysinit controls what is started at bootup. It runs all the scripts under the rc.d directory depending on which runlevel is invoked and then runs rc.rofilesys. However, rc.local which normally is the last script it runs is commented out and won't be run. You can make it run rc.local by uncommenting that section:

```
cd /etc/rc.d
if [ -f "./rc.local" ];then
  if [ -x "./rc.local" ];then
    echo "*** Running rc.local"
    ./rc.local
  fi
fi
```

Alternatively, you could also modify /home/QtPalmtop/qpe.sh and make it run /etc/rc.d/rc.local

```
    sdisp /home/QtPalmtop/pics144/Startup_screen.bmp &
    /etc/rc.d/rc.d/rc.local
    cd
    if [ -f /etc/restorefile ]; then
        export LD_LIBRARY_PATH=/usr/QtPalmtop.rom/lib
    else
        export LD_LIBRARY_PATH=$QTDIR/lib
    fi
    qectrl -c
```

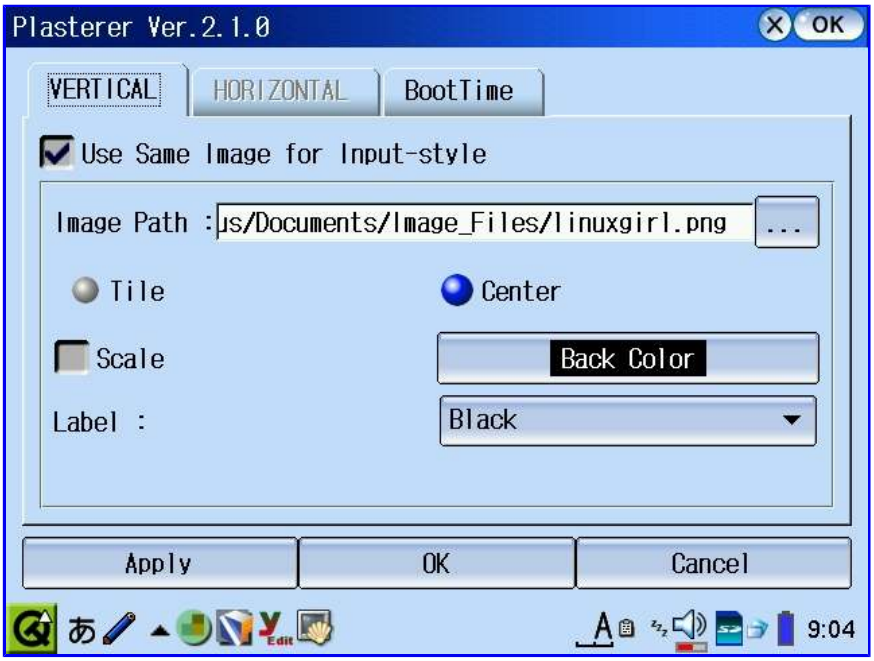
The difference between running rc.local from rc.sysinit and qpe.sh is that the output of the scripts executed from rc.local will be displayed during the boot process if its run from rc.sysinit (if using tetsu's kernel) while nothing will be displayed when running it from qpe.sh because it will be covered by the boot screen image.

Wallpaper Customisation

The C3000 and C3100 allow you to use a wallpaper which can be a png or jpg file. By default, if you use the Appearance application under the Settings tab, your selected image will be tiled unless you pick an image that exactly fits the screen (640x420 or 480x620 minus the Qtbar) depending on which your preferred orientation is. However, if you rotate the screen, then the image will be out of proportion and will be tiled.

There is a way around this if you install Plasterer [plasterer_2.1.0-1_arm.ipk] which allows you to choose an image that you can center or tile. It will then generate two images for you, **vImage.png** and

hImage.png which will be used as your wallpaper depending on which orientation you are in. You can replace those two files with your preferred images which can be two completely different images.



I have created the following background images with their horizontal and vertical pairs so they look decent on the Zaurus in either orientation:

linuxgirl 11linux mypda thinklinux

tuxback unix2000 tuxdrinkxp thinkshell

Note: Most of the above images were taken from Cresho's zaurusthemes.org site. There are plenty more pretty background images over there. These are just my favourite ones.

You will notice that this application is in Japanese. Use the Menu Language Switcher (en) tool under the Settings tab (if you have **langswitch** installed) to move the qm file and it will be in English.

Once you have used Plasterer to generate hImage.png and vImage.png you can use bgswitch [bgswitch_0.1_arm.ipk] to switch between your available backgrounds provided they are located in /home/zaurus/Documents/Image_Files/wallpaper

ScreenSaver Customisation

The default screensaver on the C3000 and C3100 is rather ugly. You can replace it with something much better. To do that install the following:

- LUSScreenSaver - [LUSScreenSaver_1.4.6-1_arm.ipk]
- LUSScreenSaverUtil - [LUSScreenSaverUtil_1.3.5-1_arm.ipk]

In addition, also install at least one of the following:

- LUSSSFish - [LUSSSFish_1.0.0-1_arm.ipk]
- LUSSSMessage - [LUSSSMessage_1.0.0-1_arm.ipk]
- LUSSSPicture - [LUSSSPicture_1.1.1-1_arm.ipk]
- LUSSSUniverse - [LUSSSUniverse_1.0.0-1_arm.ipk]

You will notice that this application is in Japanese. Use the Menu Language Switcher (en) tool under the Settings tab (if you have **langswitch** installed) to move the qm file and it will be in English.

Menu and Tab Customisation

You can re-organise tabs and applications using the **TabConf/TabSetting** tool. Alternatively, you can also move files around under /home/QtPalmtop/apps, but those changes will not apply until you restart Qtopia, reboot the Z, or run TabSetting and apply the settings by saving them. There are also two hidden files in each sub-directory called .order and .directory which can be manually changed as well.

In addition, you can customise the icons that are displayed for each application on the menu, and also the icons displayed for the applications according to their mime associations.

The icons for the menu and the tabs are located under /opt/QtPalmtop/pics144. Qtopia will always search this location for the icons first. If no icon is found with the specified name, then it will also search under /opt/QtPalmtop/pics as well. If an icon exists with the same name in each location, then the icon in /opt/QtPalmtop/pics144 is used.

The default icons that come with the default Sharp ROM on the C3000 are located under /hdd1/usr/QtPalmtop.rom/pics144 and /hdd1/usr/QtPalmtop.rom/pics and are symbolically linked to /opt/QtPalmtop/pics144 and /opt/QtPalmtop/pics144 respectively.

Similarly on the C3100, they are located under /usr/QtPalmtop.rom/pics144 and /usr/QtPalmtop.rom/pics and are symbolically linked to /opt/QtPalmtop/pics144 and /opt/QtPalmtop/pics144 respectively.

The icons for file association are by default searched under /opt/QtPalmtop/pics. When installing applications, they add their icons to /opt/QtPalmtop/pics144 and/or /opt/QtPalmtop/pics. Usually icons under /opt/QtPalmtop/pics144 are 64x64 in size and icons under /opt/QtPalmtop/pics are 32x32 in size. However, some special smaller icons of size 28x28 are also located in /opt/QtPalmtop/pics144 and size 14x14 under /opt/QtPalmtop/pics

The icons can be customised by replacing the icons used by the applications or copying additional icons to those locations and using the TabConf utility to change the icons used for the applications. Alternatively, each .desktop file under /opt/QtPalmtop/apps can also be manually modified. If you replace an icon by copying over another icon file with the same name, then you will not see the new icon until you reboot because Qtopia still has the old icon in cache.

customised start menu:

The following is a list of the names of the icons that need to be replaced in order to change the look and feel of the default Qtopia desktop. Since some of these icons are symbolic links, the symlinks need to be deleted first before a new icon can be copied to replace them. I have replaced and added the following default icons:

Settings

- **Installer Icon** - qinstall_icn.png
- **TabSetting Icon** - tabconf.png
- **Light Icon** - Light.png
- **Sound Icon** - ssoundconf.png
- **Appearance Icon** - Appearance.png
- **Network Icon** - PPPConnect.png
- **Security Icon** - Security.png
- **Calibrate Icon** - Calibrate.png
- **ApplicationKeys Icon** - CustomizeKeys2.png
- **SystemTime** - DateTime.png
- **UserDic Icon** - userdic.png

- **Backup/Restore Icon** - BackupRestore.png
- **SystemInfo Icon** - SystemInfo.png
- **Migration** - DataMoving.png
- **ReceiveData** - DataMovingSL.png
- **PC-Link(Samba) Icon** - qtsamba.png
- **IR-Receive Icon** - Infrared.png

Applications

- **TextEditor Icon** - TextEditor.png
- **Calendar Icon** - DateBook.png
- **AddressBook Icon** - AddressBook.png
- **TodoList Icon** - TodoList.png
- **Email Icon** - EMail.png
- **MoviePlayer Icon** - MPEGPlayer.png (pics only)
- **HancomSheet Icon** - hancomsheet.png
- **HancomWord Icon** - hancomword.png
- **ImagePad Icon** - zimager/zimager.png
- **Dictionary Icon** - Zten.png (pics only)
- **Translator Icon** - translator.png (pics only)
- **Calculator Icon** - Calculator.png
- **CityTime Icon** - CityTime.png
- **Clock Icon** - Clock.png
- **HelpBrowser Icon** - HelpBrowser.png
- **MusicPlayer Icon** - MusicPlayer.png
- **NetFront Icon** - nf_logo.png (pics only)
- **PhotoStorage Icon** - photostorage.png (pics only)

Menu and File Manager:

- **Start Button** - go.png (pics144 only)
- **Volume Button** - volume.png (pics144 only)
- **Wait Button** - wait.png (pics144 only)
- **ZoomIn Icon** - zoomin.png (pics144 only)
- **ZoomOut** - zoomout.png (pics144 only)
- **Rotation Icon** - transform.png (pics144 only)
- **Restart Icon** - Restart.png
- **Shutdown Icon** - Shutdown.png
- **Zaurus Home Icon** - myzaurus.png
- **Main Icon** - MainDevice.png
- **Main Icon (small)** - MainDeviceS.png
- **Folder Icon** - slfolder.png
- **Folder Icon (large)** - slfolder_l.png
- **Docs Icons** - DocsIcon.png (pics144 only)
- **Settings Icon** - SettingsIcon.png
- **Apps Icon** - AppsIcon.png
- **Games Icon** - Games.png

Additionally, more icons were added, some are listed below

Menu Tabs:

- **AppsTab Icon** - mbfolder-util.png
- **OfficeTab Icon** - mbfolder_office.png
- **MultimediaTab Icon** - mbfolder_multimedia.png
- **GamesTab Icon** - mbfolder_games.png
- **SettingsTab Icon** - mbfolder_system.png
- **GraphicsTab Icon** - mbfolder_graphics.png
- **JavaTab** - java.png

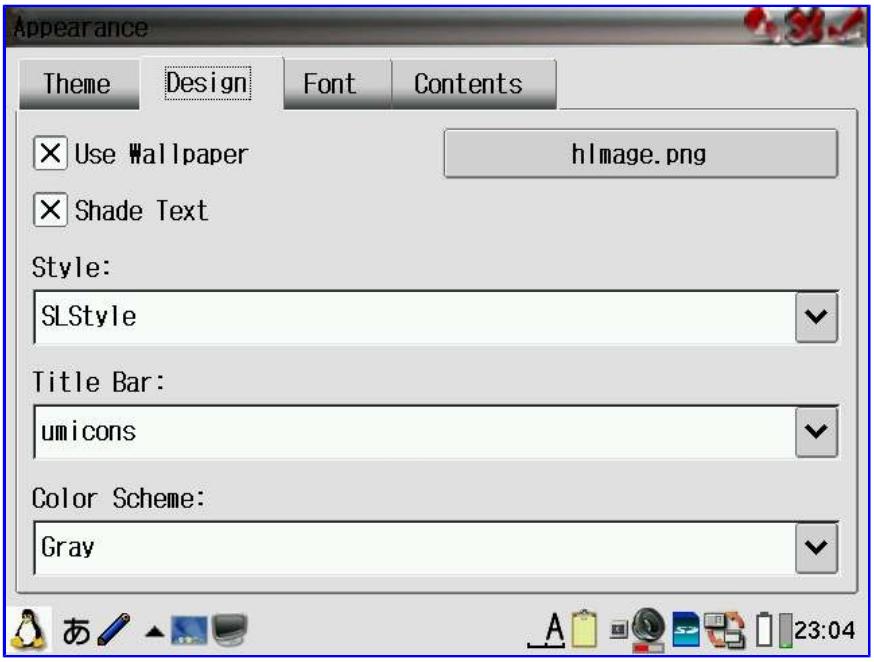
TaskBar Applets:

- **Suspend Icon** - tb_suspend.png
- **Clipboard Icon** - tb_clipboard.png
- **TaskList Icon** - tb_tasklist.png

Misc:

- **Run Icon** - exec.png
- **Aim Icon** - aim.png
- **Mirc Icon** - mirc.png
- **MSN Icon** - msn.png
- **Yahoo Icon** - yahoo.png
- **Apple Icon** - apple.png
- **BSD Icon** - bsdunix.png
- **Mac Icon** - mac.png
- **Microsoft Icon** - microsoft.png
- **Sun Icon** - sun.png
- **Book Icon** - book.png
- **Bunko Icon** - bunko2.png
- **Controller Icon** - controller.png
- **Database Icon** - database2.png
- **Download Icon** - download.png
- **FileServer Icon** - fileserver.png
- **Gens Icon** - gens.png
- **Hd Icon** - hd.png
- **InputConfig Icon** - inputconfig.png
- **Kino Icon** - kino.png
- **Konsole Icon** - konsole.png
- **LanConfig Icon** - lanconfig.png
- **Laptop Icon** - laptop.png
- **Microphone Icon** - microphone.png
- **mySQL Icon** - mysql.png
- **Paint Icon** - paint.png
- **Screen Icon** - screen.png
- **Server Icon** - server.png
- **Star Icon** - star.png
- **SMBController Icon** - smbcontroller.png
- **SMBMounter Icon** - smbmounter.png
- **Teleport Icon** - teleport.png
- **USB Icon** - usb.png
- **USBDrive Icon** - usbdrive.png

The zicons-wmtux package [zicons-wmtux_0.2_arm.ipk] will replace the above listed icons with a set of nicer looking ones and also adds a few extra icons for applications. This package is based on the cool-icons package from cacko.biz and also the z-oslinux theme from zaurusthemes.org



The Appearance tool under the Settings tab can be used to switch themes. Crystal Blue is one of my favourites, although a combination of SLStyle for the Style, umicons for the Title Bar and Gray for the Colour Scheme does look very good too.

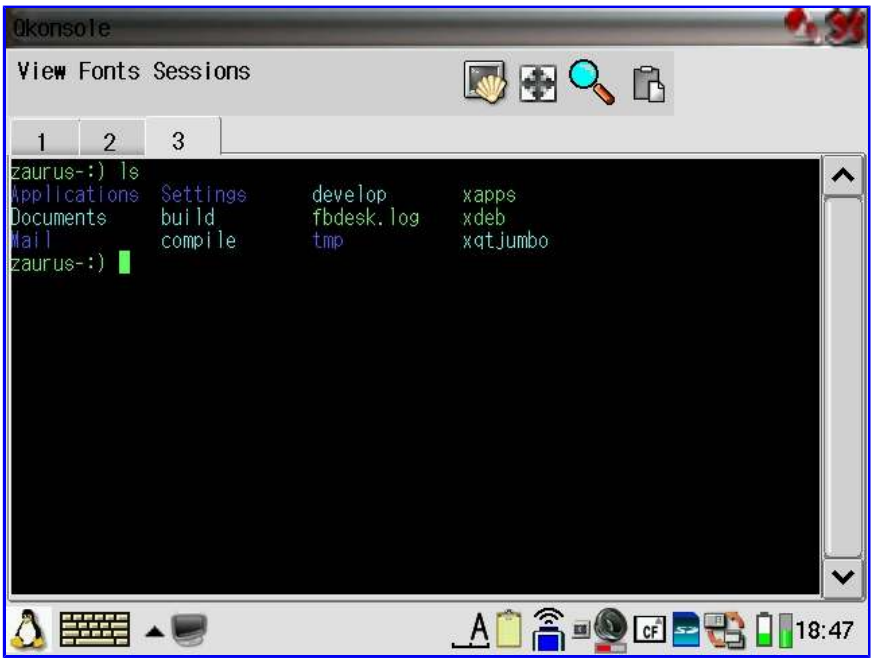
Configuring bash

The Zaurus comes with bash 2.05 by default. It also comes with ash and sh which is the default shell for the root and zaurus user accounts. The zaurus user forks off and launches bash when you launch a console from within Qtopia. When you connect to the zaurus via telnet for example or su to root, you will get the default shell (/bin/sh) which is not bash. If you want to make bash your default shell, then edit /etc/passwd and change /bin/sh to /bin/bash

```
root:x:0:0:root:/home/root:/bin/bash
zaurus:x:500:500:Zaurus User:/home/zaurus:/bin/bash
```

You can also customise the bash prompt by creating or editing a file called .profile or .bashrc which is in the user's home directory, eg: /home/zaurus/.profile or /home/root/.profile

You can change the bash prompt by changing the PS1 variable definition to something simple like **zaurus-:)** or even something fancy with different colours.



If you don't have a .bashrc file, you can just copy .profile for starters and then modify it.

Zaurus Home Directory

The zaurus user's home directory (/home/zaurus) is located on the /home partition (/dev/mtdblock3). On the C3000 it is only 4MB in size. It is physically located on the internal flash memory which is only 16MB in total on the C3000. This partition cannot be made bigger since the rest of the flash memory is used for the main rootfs (/dev/mtdblock2), the emergency rootfs (/dev/mtdblock1). The /home partition is also used to store kernel modules (/home/root/modules), system configuration (/home/zaurus/Settings and /home/etc/) and other things. In addition, some applications also store their configuration and data into the zaurus home directory (this is what linux apps are supposed to do), but it is not so ideal on the C3000. Thus the /home partition quite quickly fills up. It it advised to not store large files there, in fact, try avoiding saving anything there to preserve the precious space available on /home.

If you do run out of space, you can move some files and directories to /hdd2 or /hdd3 and symlink the files/directories. To simplify this, I have created a script zhomefix which will move all files and directories in /home/zaurus starting with . to /hdd2/zaurushome and symlink them back. If you are low on space on /hdd2 as well, you can modify zhomefix to move files to /hdd3/zaurushome instead.

On the C3100, however, the situation is slightly different. The internal flash memory on the C3100 is 128MB in size compared to the tiny 16MB on the C3000. Now you would think that the C3100 won't run out of space so easily. Unfortunately, that is not the case. The rootfs is now allocated 32MB instead of 4MB which was what it was on the C3000, and remember, there is also the emergency rootfs (smf). So

after allocating some of the space for the other stuff, there is still 89MB of space left on /home. This isn't so bad after all you think, but wait, there is another surprise. On the C3000, the internal MicroDrive was partitioned into /hdd1, /hdd2 and /hdd3. The default binaries and settings were stored on /hdd1 which was a read-only filesystem. All the applications were installed to /hdd2 and the remaining /hdd3 was used for data. On the C3100, the content of /hdd1 has moved to the rootfs and /hdd2 has moved to /home. Those two partitions (/hdd1 and /hdd2) are now about 9MB in size each on the C3100 and are more or less empty and not in use. This makes /hdd3 much bigger on the C3100 compared to the C3000.

However, this in effect makes /home on C3100 equivalent to /hdd2 and /home on the C3000. There was around 400MB allocated to /hdd2 on the C3000 for installing applications. We only have 89MB on the C3100. Luckily, this 89MB is located on a jffs2 filesystem which has built-in compression so we could be able to install around 200-300MB of applications and stuff if we are lucky. Still, we will run out of space eventually. If that happens, we could uninstall some applications (such as the additional applications that came with the C3100 but weren't present on the C3000) or we could move some files to /hdd3, but be careful, /hdd3 is by default a FAT filesystem and does not support symbolic links which some applications might require. See the Filesystem section on discussion on using cramfs or ext2 loopback filesystem on /hdd3 to extend the amount of installable space. Alternatively, you can install to SD or CF card, but some applications require to be installed to non-FAT partitions so you might need to reformat them.

HDD3 Considerations

Both the C3000 and C3100 contain an internal 4GB MicroDrive which is partitioned into /hdd1, /hdd2 and /hdd3. By default, /hdd3 is formatted as a FAT filesystem so it can be shared as a USB drive when the Zaurus is connected to a computer as a USB slave and accessed from Windows. The first partition, /hdd1 is a read-only partition containing the OS images and binaries required to run and restore the Z to factory default on the C3000. On the C3100, it is an almost empty partition with a size of 9MB. It contains a file called hdimage2.tgz that has the directory structure and sample template files for hdd3. The second partition, /hdd2 is where applications get installed to on the C3000. On the C3100, it is also an empty partition with a size of 9MB. The third partition, /hdd3 is allocated the remainder of the MicroDrive and is a FAT formatted partition. It can be used to store data on files of any type, including relatively large files. Alternatively, /hdd3 could potentially also be used for applications when /hdd2 or /home is full if you apply a few modifications.

However, since the default /hdd3 is using the FAT filesystem, no symlinks can be created on it and file permissions and ownership are also not available. Thus, there are a few drawbacks with having /hdd3 as a FAT partition. You can either reformat the entire /hdd3 to linux filesystem (ext2/ext3) or repartition /hdd3 into /hdd3 and /hdd4. This leaves a smaller FAT formatted /hdd3 and additionally a /hdd4 with a linux partition.

Before repartitioning /hdd3, make sure you backup everything on it first. The dictionary files (dict1 and dict2) which are by default on /hdd3 can be found on the first two CD-ROMs that came with the Zaurus. The C3100 has a third CD-ROM which contains the files for the Contents_Files directory. The contents of the sd_map directory (C3100 only) are located under a sub-directory under the Applications directory on the first CD.

/hdd3 is usally mounted from /dev/hda3 or /dev/hdc3 depending on how you booted. If you booted with no CF card, then it will be /dev/hda3, however, if you had booted with a CF card inserted, then it will be /dev/hdc3.

The steps required for splitting hdd3 into two are:

- unmount hdd3
- run fdisk and delete hdd3
- run fdisk and create hdd3 as FAT
- run fdisk and create hdd4 as EXT2 or EXT3
- format hdd3 as FAT
- format hdd4 as EXT2 or EXT3

There is also a tool called **parted** which allows you to resize your existing partitions without having to remove them. This is certainly very useful tool, but remember to backup your hdd3 before resizing it if you have files on hdd3 that you want to keep. Although parted can resize the partition without wiping your data, it is not guaranteed. There may be instances where resizing could corrupt the partition so its always wise to do a backup first. Also run fsck after using parted to verify that the partition has been resized successfully without corruption.

Once that is done you need to remount / as read/write and create a mount point for hdd4, ie /hdd4. Remember to remount / to read-only after you have created your mount point. Also you will need to create a startup script to mount hdd4 during bootup. But be careful since hdd4 can boot up as /dev/hda4 or /dev/hdc4 depending on whether a CF card is inserted during bootup or not. My automounter package [automounter-c3000_0.5.0_arm.ipk] will automatically mount hdd4 if it detects it.

Alternatively, if you don't need your Zaurus to act like a USB drive, or your PC runs Linux, then you could just reformat the entire /hdd3 to linux filesystem.

You might also consider creating a small swap partition while you are at it. A swap partition is faster than a swapfile. A swap partition between 64MB and 256MB should be fine depending on your usage and applications.

Also note that the default installer tool (qinstaller) won't let you install applications to either /hdd3 or /hdd4. ipkg will allow you to install to those locations but won't relink the applications for you, so you will have to use ipkg-link afterwards which is not included with the default Sharp ROM. My xipk script which is part of my ipktools package enables you to install to /hdd3 and/or /hdd4, and also relinks the files and directories for you. In addition, it uses the same mechanism as qinstaller and thus applications installed with xipk can be uninstalled using the qinstaller.

If you want to maximise the space on hdd3 and you don't care about the Japane se/English dictionary and translator, then you could remove the dictionary files under /hdd3/dict1 and /hdd3/dict2. If you later decide that you do want them, simply copy them back from the CD-ROM (so don't loose your CD-ROMs). On the C3100, there is additionally the MobileMap application which has some files under /hdd3/Documents/sd_map. You can uninstall the application and remove the sd_map directory to get more space. You can re-install MobileMap from the first CD, and find the contents of sd_map under X:/Applications/MobileMapData/sd_map. The MobileMapData part is in katakana. There is also the Contents_Files directory containing many Japanese books and reading material on the C3100. If you don't know Japanese, you probably want to hide the Contents tab. This can be done though the Appearance tool under the Settings tab. You probably also want to remove the /hdd3/Documents/Contents_Files directory afterwards as well. If you ever want it back, you can simply copy it from the third CD.

Lastly, /hdd3 gets wiped when you do a factory reset, but you can disable that behaviour. To do that you need to first remount / as rw and then modify /root/etc/rc.d/rc.rofilesys and comment out the following section:

```
if [ "$HDDCLEAR" = "YES" ]; then
       dd if=/dev/zero of=/dev/${IDE1}3 > /dev/null 2> /dev/nulls
fi
mkfs.vfat -F 32 /dev/${IDE1}3 2> /dev/null > /dev/null
```

Of course, after you do a factory reset, you will need to fix rc.rofilesys again so the next time you do a reset it won't wipe your hdd3. Alternatively, you could also update .home_default.tar and replace the rc.rofilesys in there with the hacked version and not worry about it anymore.

On the C3100, /hdd1 and /hdd2 doesn't contain anything important and are a waste of space because those partitions are not really used except for factory reset to wipe /hdd3 which sux anyway. Thus you can hack rc.rofilesys to not even mount them or you could resize the partitions so that /hdd1 is a swap partition, /hdd2 is a linux filesystem (instead of /hdd4) and /hdd3 becomes a smaller fat partition.

Here is the default partition table:

```
Disk /dev/hda: 4095 MB, 4095737856 bytes
16 heads, 63 sectors/track, 7936 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Device Start End Blocks ID System
/dev/hda1 1 20 10048+ 83 Linux
/dev/hda2 21 40 10080 83 Linux
/dev/hda3 41 7936 3979584 c Win95 FAT32 (LBA)
```

Here is my custom partition table which has /hdd1 as a 256MB swap partition, a 1.2GB ext3 partition on /hdd2 and a 2.5GB on /hdd3:

```
Disk /dev/hda: 4095 MB, 4095737856 bytes
16 heads, 63 sectors/track, 7936 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Device Boot Start End Blocks Id System
/dev/hda1 1 256 128992+ 82 Linux swap
/dev/hda2 257 2790 1277136 83 Linux
/dev/hda3 2791 7935 2593080 c Win95 FAT32 (LBA)
```

In order to do this, you need to replace /root/etc/rc.d/rc.rofilesys after you have remounted / as rw. You also need to remove the NotAvailable file from the unmounted /hdd1 and /hdd2.

I have successfully used parted to shrink /hdd3 dynamically without destroying my existing files but backing up the files is still recommended just in case. Then I used fdisk to recreate /hdd1 and /hdd2 with their new sizes and formatted them using mkswap and mk2fs -j respectively.

Now when the C3100 boots up, it automatically mounts /hdd3 as before but it won't erase /hdd3 on a factory reset, and it also mounts /hdd2 as ext3 if it exists. Additionally, it also enables the 256MB swap partition (formerly /hdd1) and mounts tmpfs as 2MB instead of 1MB.

MicroDrive Performance

The C3000 and the C3100 both have a 4 GB CF MicroDrive internally, which is used as its harddisk to store data. Since a CF Flash card is generally faster than a MicroDrive, it would make the Zaurus faster if the CF MicroDrive is replaced with a CF Flash card. This makes sense for the C3000 where the applications and binaries are stored on the MicroDrive and there is a slight delay for the MicroDrive to spin up when it has gone into powersaving mode after some time of disk inactivity. However, for the C3100 it would not make such a big impact on application performance since they actually are on the flash memory instead of the MicroDrive. Still, if the application is also stored on the MicroDrive, ie /hdd3 is used for installing additional applications such as PocketWorkstation and a large swapfile, then it would make sense too.

To do this, you would need a CF Flash card to replace the MicroDrive with. Make sure the Flash card you are using has a similar capacity to the MicroDrive (4GB would be good, 2 GB is managable and bigger ones should be better). However, make sure the Flash card you are using is faster than the current MicroDrive inside the Zaurus (the Z has a 4GB Hitachi MicroDrive inside), otherwise you won't be gaining anything.

What you want to do is mirror the 4GB MicroDrive to the Flash card. You can d o this by inserting the Flash into the CF slot on the Zaurus and partition it ex actly like the MicroDrive using **fdisk**. Then use the **dd** command to copy each partition, ie hdd1, hdd2 and hdd3. Unmount hdd2 and hdd3 before you copy them or remount them to read-only. The other partitions are located in the internal flash memory (also called Nand). Alternatively, you can also use **parted** to copy the entire partition from one disk to another. Once this is done, you can open up your Z and swap the two drives. This will void your warranty, so make sure you understand and know what you are doing. It is your own responsibility if you break your Zaurus or any parts of it. Finding replacement parts will be extremely difficult unless you live in Japan, so be careful and consider the consequences of your actions, or modifications. Let me say it again. If you open your Zaurus up to replace parts, you void your warranty. If you break something during the process, then you are on your own since you just voided your warranty.

Zaurus Backup

You should always backup your system since that is the only way to recover if something goes wrong.

The C3000 and C3100 come with a backup and restore tool which is located under the Settings tab. Use this application to backup your Zaurus. It allows you to backup your system (flash, applications and configurations) to either SD, CF or /hdd3. Basically, everything except /hdd3, /hdd1, /root and /mnt will be backed up and can be restored which means you will need to backup /hdd3 by other means. Before you backup, make sure you unmount any loop devices that are mounted, unless they are mounted under /mnt, otherwise they will be backed up as well (which you think might be great) but you won't be able to restore the backup image (because the additionally backed up files on the loop device(s) will make the backup image bigger than the backed up partitions). If you have automounter installed, you can unmount all the loop devices by running the following:

```
# su
# automounter stop
```

Once you have backed up all the files on your Zaurus, you can remount all the loop devices by running the following:

```
# su
# automounter start
```

Since /hdd3 is quite large, you either need to get a big CF card, or mount a Samba or USB drive that has enough space to hold your data. A USB drive would be the best (cheaper than CF drive and faster than Samba since its connected directly and not over a network unless you have got a fast network).

Assuming you have your USB disk mounted as /mnt/usbdisk1 you could do the following to backup /hdd3:

```
# tar cf - /hdd3 | gzip - > /mnt/usbdisk1/hdd3-backup.tgz
```

If you are paranoid you can backup /hdd1, /hdd2 and /home as well:

```
# tar cf - /hdd1 | gzip - > /mnt/usbdisk1/hdd1-backup.tgz
# tar cf - /hdd2 | gzip - > /mnt/usbdisk1/hdd2-backup.tgz
# tar cf - /home | gzip - > /mnt/usbdisk1/home-backup.tgz
# tar cf - /root | gzip - > /mnt/usbdisk1/root-backup.tgz
# tar cf - /mnt | gzip - > /mnt/usbdisk1/mnt-backup.tgz
```

The Zaurus backup tool basically shuts down Qtopia so all open files are closed and then tars up /hdd2 and /home which become the backup image. This is much safer than the above approach while Qtopia is still running. You should, however, gzip the backup image to save some space. There is no real need to

backup /hdd1 each time since it is a read-only partition and does not change unless you have changed something on it manually or applied an update via flashing.

I have written a script called hdbackup which will backup /hdd3 by archiving and compressing each directory separately and datestamping them so regular backups can be made by simply running a single command.

Alternatively, you can connect your Zaurus to your Windows PC through the USB link cable and use Windows backup software or anything else you like to backup the USB drive that the Zaurus is recognised as. Since the USB PC connection is buggy, you might be better of enabling Samba and then backup /hdd3 over the network or USB cable.

Lastly,don't forget about backing up your SD and CF card also. They can get corrupted or fail without warning as well, so make sure you back them up to.

In addition, there is also a NAND backup feature in the Zaurus Diagnostic Menu which allows you to backup the entire NAND. Since the C3000 only has 16MB of NAND flash and everything actually sits on the hdd3, doing a NAND backup won't buy you much. On the C3100, however, everything is on the NAND except for the data on /hdd3, thus making a NAND backup for the C3100 gives you a reliable system image which you can use to restore your C3100 if you really mess it up. The same is not the case with the C3000 so be very careful with what you flash your C3000 with.

To do a Nand Backup, you need a CF or SD card which can hold the entire Nand. A 256MB card should be sufficient. To do a Nand Backup or Restore, do the following:

- Turn off or suspend the Z
- Unplug the Z and take out the battery
- Press and hold the D and M keys simultaneously
- Plug in the power
- The Maintenance Menu should appear in a few seconds
- Go to the third page in the menu
- Select either Nand Backup or Restore
- When finished turn off the Z
- Put battery back in and start Z

Zaurus Restoration/Recovery

If you manage to corrupt your Zaurus configuration so badly that you cannot boot it anymore or things just don't work any more, then you have several options to fix it.

- Factory Reset

  If you really messed up and just start over again without retaining anything, then just do a factory reset and the Zaurus will revert to its initial Japanese ROM state.

- Restore From Backup

  If you have a backup, you can restore your previous settings contained in your backup files. If you had to factory reset, then a restore can quickly get you to where you were before.

- Command Line Recovery

  This is by far the most advanced option. Use this to recover files that might not have been backed up yet before doing a restore and/or factory reset.

Please refer to Trisoft's C3000 manual on how to do the above. It would be a waste of my time to provide step by step instructions since they have it pretty much covered, and they even have emergency backup images for you to use in case you dont't have a system backup.

If you have booted into the console for recovery, then you are using the /dev/mtdblock1 partition. This is the emergency partition that you usually don't see. You will need to manually mount the usual partitions if you want to access them. /home is /dev/mtdblock3 and /dev/mtdblock2 is your root partition. Don't forget to unmount the partitions after you have finished your changes or they will be rolled back and all the files remain unchanged.

Zaurus ROM Update

For the C3000, you should update your Sharp ROM to 1.11JP if you are still on 1.01JP

There is no updated Sharp ROM for the C3100 yet.

- Put card_update_3000111.exe onto a Windows machine (you got one of those right?) and run it to extract the files. (it's a self extracting zip file).
- Then copy all the extracted files to a SD or CF card (if your windows box has no card reader, then put the memory card into your Zaurus and connect it via USB cable and switch it to share your card instead of your hdd so you can transfer the files directly to the card). Make sure you put the files into the card's root directory, ie don't put them into any folder and don't copy the folder they were in.
- The following files should be on the root of your memory card:
  - initrd.bin(about 4,337KB)
  - zImage.bin(about 1,264KB)
  - mversion.bin(about 16KB)
  - updater.sh(about 7KB)
  - hdimage1.tgz(about 19,734KB)
- Turn off your Zaurus and disconnect the power cable (and any other cables). Unlock the battery compartment and press the little reset button with your stylus. Put the lid back on and make sure to lock it again.
- Now make sure your Zaurus is plugged in to the AC power. You don't want to be on battery power and have your Zaurus run out of juice in the middle of the update process (you can kiss your Zaurus goodbye if that happens). Also make sure your card with the update files is inserted.
- The charge indicator should be orange now. Hold the "OK" key on the keyboard or the back of the Zaurus and turn on the device with the "On/Off" button.
- You should see the maintenance menu and select option 4 to update/flash the ROM. On the next screen, select either 1. CF or 2. SD depending on where your update files are. Now confirm to proceed with the update by selecting Y. Wait about 5 minutes after which your Zaurus should reboot. You're done. You should have 1.11 JP ROM now.

Warning: Only flash your Zaurus with a ROM intended for your specific model and make sure you downloaded the complete files. Never ever flash your Zaurus with a ROM for another model. It will cause you many sleepless nights trying to restore it to a working condition.

Zaurus Kernel Replacement

There are several replacement kernels for the C3000 and C3100 which enhance the stock kernel that ships with Sharp's ROM. You can even build your own if you want (and know how to). The kernel source is available on Sharp's developer website. However, there are some smart people who already build and tested their own enhanced kernels. One of those is Tetsu's special kernel which has been build for optimised speed and also includes iptables and bluetooth modules. It also makes the battery status more accurate and it has a few bug fixes too.

Warning: The kernel is an important part of the Linux OS. A bad kernel can corrupt your Zaurus, so don't play around with it unless you know what you are doing and install the correct kernel for your model.

- Download and place the files onto a CF or SD card.
- Turn off your Zaurus. (suspend it).
- Unlock the battery compartment and push the little reset button.
- Put the lid back on and lock it again.
- Plug your Zaurus into the AC power.
- Hold the "OK" key on the keyboard or the back of the Zaurus and turn on the device with the "On/Off" button.
- You should see the maintenance menu and select option 4 to update/flash the ROM.
- Select either 1. CF or 2. SD depending on where your update files are.
- Confirm to proceed with the update by selecting Y.
- Reset your Zaurus (press the little reset button inside the battery compartment and then press the "On" key).
- Install the kernel module ipk file that came with the kernel.

Zaurus Maintenance

fsck needs to be run on the Linux filesystems from time to time to check for inconsistencies in the file systems and to fix it if there are any. fsck is similar to the chkdsk or scandisk command in DOS and Windows.

Running fsck on a mounted filesystem is not recommended so the safest way to run it on the C3000 is from the maintenance menu. Unmounting the partitions on the MicroDrive is much easier on the C3100 since it runs off the flash instead of the disk. For FAT partitions, ie /hdd3, use fsck.vfat instead of fsck.

To perform fsck via the maintenance menu do the following:

- Shutdown Zaurus
- Remove battery lid and press the reset button
- Put battery lid back on and lock battery compartment
- Plug the power cable in
- Hold OK button and turn Z on
- Select Option 2 (data check)
- Select Option 2 (run fsck)
- Confirm (left option)
- Wait for fsck to finish
- Restart Zaurus (using the reset key inside the battery compartment)

The fsck from the Maintenance Menu checks all three partitions on the MicroDrive, ie /hdd1, /hdd2 and /hdd3. It does not fsck /home or /root which is located on Flash. In fact, there is currently no known way of fscking a jffs2 filesystem which /home and /root are formatted as.

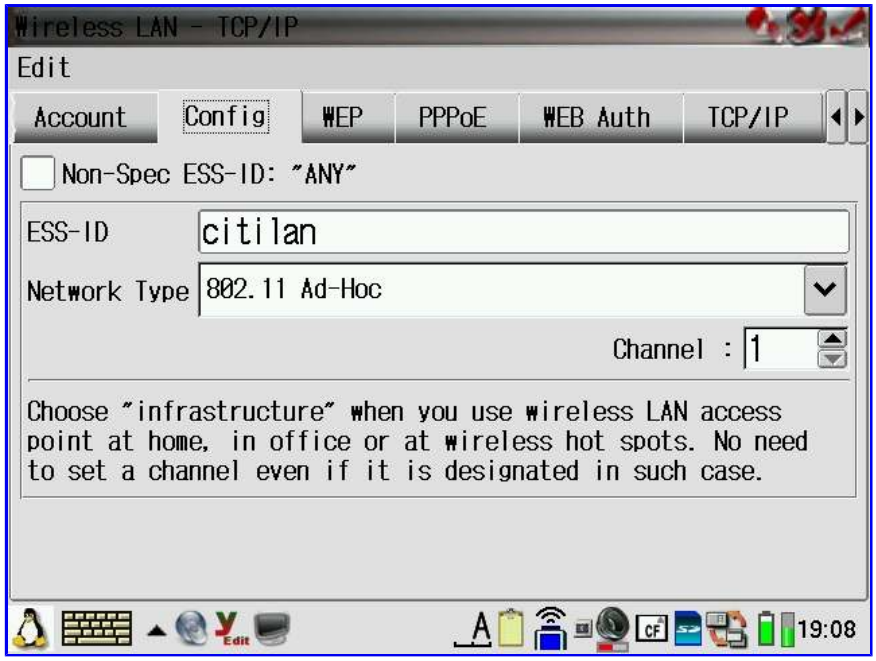
You should also regularly fsck your SD and CF card. Please unmount them before fscking them. Here is how you would fsck a FAT formatted SD card:

```
# su
# umount /mnt/card
# fsck.vfat /dev/mmcda1
# mount /mnt/card
```

Zaurus Networking

Configuring a Wireless CF adaptor:

Enabling the wireless network was amazingly easy and straightforward. Just plug in the Wireless CF card and the Zaurus automatically detects it. Then run the Network config applet and enter the network info and press connect. Voila! That's it. Way too easy. This was the case using a Netgear MA701 CF Wifi card. Not all CF Wifi cards are supported so your milage may vary depending on your card.



However, you can also manually configure the network without using the Network config applet. The config files it generates are located under /home/zaurus/Applications/Network/modules and are called WLANx.conf. You will also need to edit the corresponding entry in /etc/pcmcia/wlan-ng.opts. If for any reason the applet won't let you enter a value you want (such as space in ssid), then you can edit the mentioned config files yourself.

Enabling a USB LAN adaptor:

Now this one was a bit trickier. My USB LAN adaptor came with a Linux driver, a file called rtl8150.c and all that was required was to compile it on the Zaurus (provided you got gcc to work). Anyway, I cheated and googled for *rtl8150-1.o* and found it :)

Next I had to install this driver which was quite easy. All that was required was to drop it into the following location: */lib/modules/2.4.20/kernel/drivers/usb/net* and the hotplug mechanism in Linux would detect whenever the device was connected and enable *eth0*.

Now came the slightly harder part, ie the automatic configuration of the device. The network applet seems to only work for the CF based cards so it completely ignored eth0 because it came from the USB interface. After looking at how the usbdnet and wlan is configured by the hotplug mechanism, I extended the *net.agent* to check for eth0 as well and added *net.func* and *net.conf* to automatically configure the network once the cable was plugged in.

In addition I also wrote a script called *net* to change the stored network settings so I can easily switch between networks. The configuration for *net* are stored under */etc/sysconfig/netconf* and *net* is invoked with the name of one of the config files as the parameter. Proxy settings for the Zaurus are stored under /home/zaurus/Applications/Network/modules/Proxies.conf which *net* will automatically update depending on the config being loaded.

```
# su
# cp /home/zaurus/Documents/custom/rtl8150-1.o /lib/modules/2.4.20/kernel/drivers/usb/net
# cp /home/zaurus/Documents/custom/net /usr/bin
# cp /home/zaurus/Documents/custom/net.* /etc/hotplug
# mkdir /etc/sysconfig/netconf
# cp /home/zaurus/Documents/custom/*.conf /etc/sysconfig/netconf

for example:

# net dhcp (loads the config file dhcp.conf)
# net -gui (starts with opie shell in QTopia desktop)
# net -refresh (tells NetFront that its connected already)
# net -resume (manually force network to resume)
```

Here is a sample config file for a private network:

```
DHCPC=no
IP=192.168.1.2
NETMASK=255.255.255.0
DOMAIN=
GATEWAY=192.168.1.1
PROXY=0
PROXYHOST=
PROXYPORT=
DNS1=192.168.1.1
DNS2=
```

net also has a GUI front-end using *opie-sh*, but in order to use the GUI, *sudo* needs to be configured to allow zaurus user to change the network settings, ie. ifconfig and dhcpcd. I also created a netswitch package [netswitch_0.4_arm.ipk] which will do the above steps when installed.

Some application such as NetFront insist on doing their own connection and disconnection to the network and ignore the fact that your USB network is already connected. As a workaround **net** has a *refresh* option to reset the network status whenever those programs mess with it. Simply run the following after you have launched NetFront:

```
# net refresh
```

A lot of the USB network cards use either the rtl8150 chipset or are compatible with the pegasus driver. Drivers for both are included with netswitch, so installing the netswitch package should enable your USB LAN device in most cases. However, some network cards use other chipsets. You should be able to compile your own driver if you can find the driver source. See gcc section for further details.

Using the advanced USB sync:

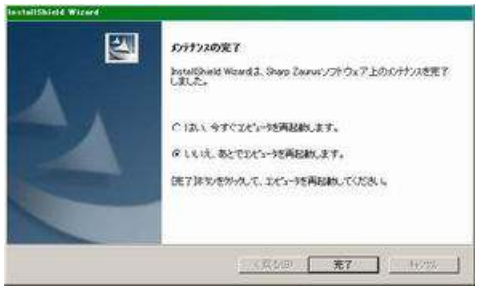
The USB sync cable which allows you to access your Zaurus as a USB disk can also be used in TCP/IP mode which means you could use that cable to network your Zaurus and your PC.

To do this, you first need to install the USB NDIS drivers onto your PC or laptop. The Zaurus Software for the PC (Windows) does not use UniCode and hence displays garbage when run on an English version of Windows (even if you have installed the Japanese language pack and your browser can display Japanese websites without problems). To fix this, change the default Windows system locale to Japanese and restart Windows. Then run X:\PCSOFT\Setup.exe from the Zaurus CD-ROM and you will be able to read the Japanese.

Select next as appropriate and use the following as guide for the options:

- Intellisync for Zaurus
- Backup/Restore
- Zaurus Shot
- Zaurus Drivers (Network)

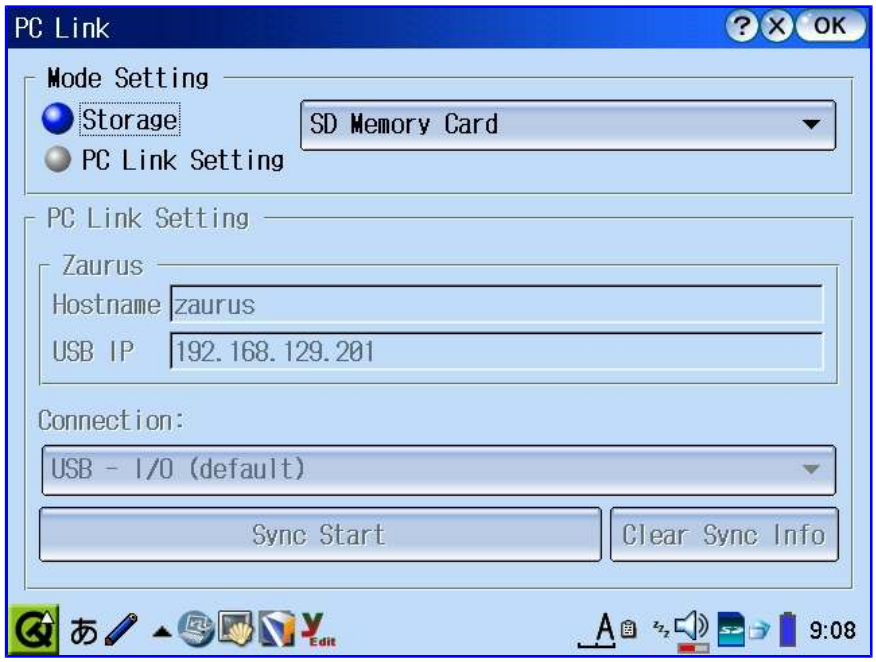
(you only need the last option for the USB network)



Restart Windows when the install has finished.

The USB network driver is now installed and **Zaurus Manager** should had started automatically. If it has not, start it manually by running "C:\Program Files\Common Files\Sharp\SL\SSPCLINK2\ComSet.exe". You should also make a copy of C:\Program Files\Sharp Zaurus 2\drivers, best is to zip up the whole directory. This is the Zaurus USB network driver. You can use this driver to re-install the USB network or install it to another machine without having to run through the whole setup process again.

Rather than carrying a floppy, CD-ROM or another USB stick around that contains the Zaurus NDIS USB driver, you could put it on /hdd3/Documents on the Zaurus itself. When you connect your Zaurus to a computer which has not been setup with the Zaurus NDIS USB driver before, it will detect the Zaurus as a plain USB storage device and allow you to copy the driver from the Zaurus. Then once you installed the driver to the computer, it will believe that it is connected to the Zaurus via a network instead of treating the Zaurus as a dumb USB disk (provided the Zaurus is in advanced USB mode).



Now, on to the Zaurus side of the configuration. Run the PC Link tool from the Settings tab and select **PC Link Setting**, then select **Connection USB-TCP/IP (advanced)**. Now just connect the USB cable (USB mini-B into Zaurus, USB A into Laptop or PC). The Windows machine should detect a new device at this point, a SL series Ver3 (NDIS 5) network adaptor, and you should be able to configure it. By default, the Zaurus would be assigned an IP address of 192.168.129.201. Assign an IP address in the same range to this new network adaptor, eg 192.168.129.101. You should now be able to ping both ways unless you have a firewall blocking it or DDE service is not enabled. If you want the Zaurus to be able to access the

internet as well, you could enable internet sharing on your Windows PC (assuming it has internet connectivity and you trust Microsoft security). If you do that, Windows will reset the IP address of your Zaurus NDIS driver to 192.168.0.1 but you can change it back to whatever value you had given it before, ie 192.168.129.101

On the Zaurus side, you need to run the following commands to setup a route to your windows box:

```
# su
# route add -host 192.168.129.101 usbd0
# route delete -net 192.168.129.0/24 usbd0
# route add default gw 192.168.129.101
```

Now that the route is configured, you should be able to ping servers by their IP addresses. In order to resolve the hostnames, you need to configure /etc/resolv.conf on your Zaurus with the DNS that is used on your Windows box. Assuming your DNS is 192.168.10.1, do the following:

```
# echo "nameserver 192.168.10.1" > /etc/resolv.conf
```

You can also automate the above on the Zaurus by modifying /etc/hotplug/usbd.func and adding the following to the end of the usbd_net_if_up() function

```
if [ "$DHCPC" = "no" ]; then
  GATEWAY=192.168.129.101
  DNS=192.168.10.1
  route add -host $GATEWAY usbd0
  route delete -net `echo $GATEWAY|cut -d. -f1,2,3`.0/24 usbd0
  route delete default
  route add default gw $GATEWAY
  echo "nameserver $DNS" > /etc/resolv.conf
fi
```

Using IrDa for networking:

Since the Zaurus has an infra-red port (IrDA), you can use it for networking as well provided you also have an IrDA port on your PC or laptop that you can configure to use PPP over IrDA (IrCOMM or IrNet). This method of networking your Zaurus would give you the slowest network speed and you usually would not use it if the other options were available to you. But if your CF slot and USB port are tied up with other things, then using IrDA for networking might be something viable.

For this to work, you would need to first choose whether to use IrCOMM or IrNet drivers. Then you would need to make sure the chosen driver is enabled on both your Zaurus and your PC or Laptop. For IrDA connectivity, one machine has to be the host and the other the client. I will describe how to make Zaurus the host and the other PC or Laptop the client. The roles can also be easily reversed. I also did not bother with security since both machines would have to be physically in close range to each other in order for this to work.

IrCOMM

The IrCOMM driver is by default already installed on the Zaurus and most Linux machines that have IrDA enabled. However, you would need to install a driver for Windows. On Windows 2000, for example, you will need to disable Image Transfer and install an IrCOMM driver (IrCOM2k). The following site describes how to setup IrCOMM on Windows2000: http://www.stud.uni-hannover.de/~kiszka/IrCOMM2k/English/manual.html. Once you have installed the IrCOMM driver, you can setup a new network connection (Direct Connection) using IrDA as the device and setting up Windows as the client with no password prompt.

On the Zaurus, you will need to do the following to make it start IrDA as the host:

```
# su
# /etc/rc.d/init.d/irda start
# pppd /dev/ircomm 9600 10.10.10.21:10.10.10.20 local noauth nodetach
```

On the IrDA client, if you are running Linux (and IrCOMM is already setup), you can simply do the following:

```
# su
# /etc/rc.d/init.d/irda start
# pppd /dev/ircomm 9600 nodetach
```

If you are running Windows (and you have setup the direct connection using IrDA as client) you can simply double click on the Direct Connection icon to connect.

Once they are paired successfully, you can ping the other box from the Zaurus as 10.10.10.20. The Zaurus would be 10.10.10.21 in this example.

To stop IrCOMM, simply press **Fn + c** and then run **/etc/rc.d/init.d/irda stop**

IrNet

You will need the IrNet module on the Zaurus and your other Linux box. However, it is already installed on Windows 2000 so all you need to do on Windows is to create a new Direct Connection using IrDA interface.

To install and enable IrNet on the Zaurus, you will need to copy irnet.o to /lib/modules/2.4.20/kernel/net/irda/irnet.

```
# su
# mkdir -p /lib/modules/2.4.20/kernel/net/irda/irnet
# cp irnet.o /lib/modules/2.4.20/kernel/net/irda/irnet
# mknod /dev/irnet c 10 187
# chown root:root /dev/irnet
# chmod 644 /dev/irnet
# insmod irnet
# /etc/rc.d/init.d/irda start
# pppd /dev/irnet 9600 10.10.10.21:10.10.10.20 local noauth nodetach
```

On the IrDA client, if you are running Linux (and IrNet is already setup), you can simply do the following:

```
# su
# /etc/rc.d/init.d/irda start
# pppd /dev/irnet 9600 nodetach
```

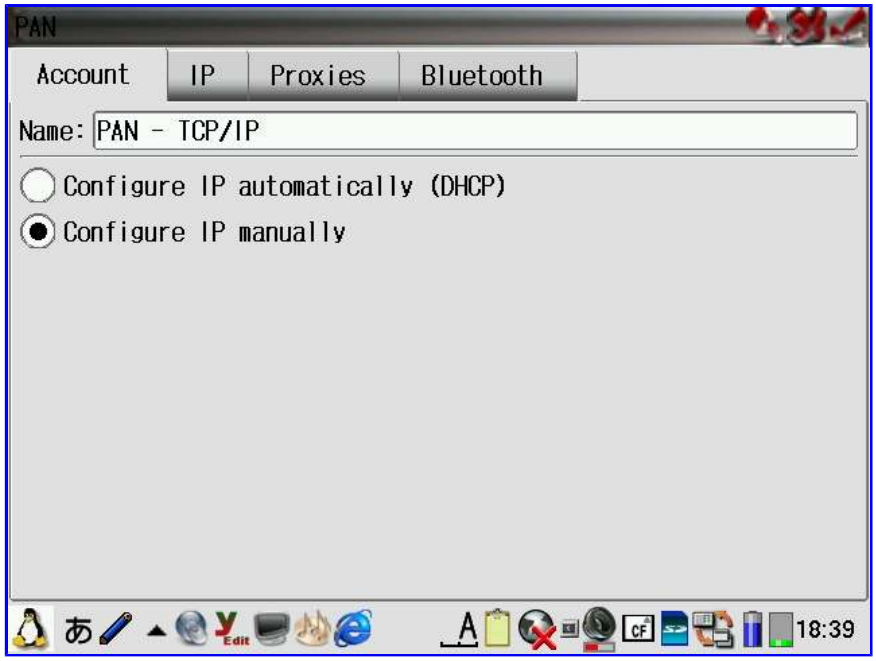
If you are running Windows (and you have setup the direct connection using IrDA as client) you can simply double click on the Direct Connection icon to connect.

Once they are paired successfully, you can ping the other box from the Zaurus as 10.10.10.20. The Zaurus would be 10.10.10.21 in this example.

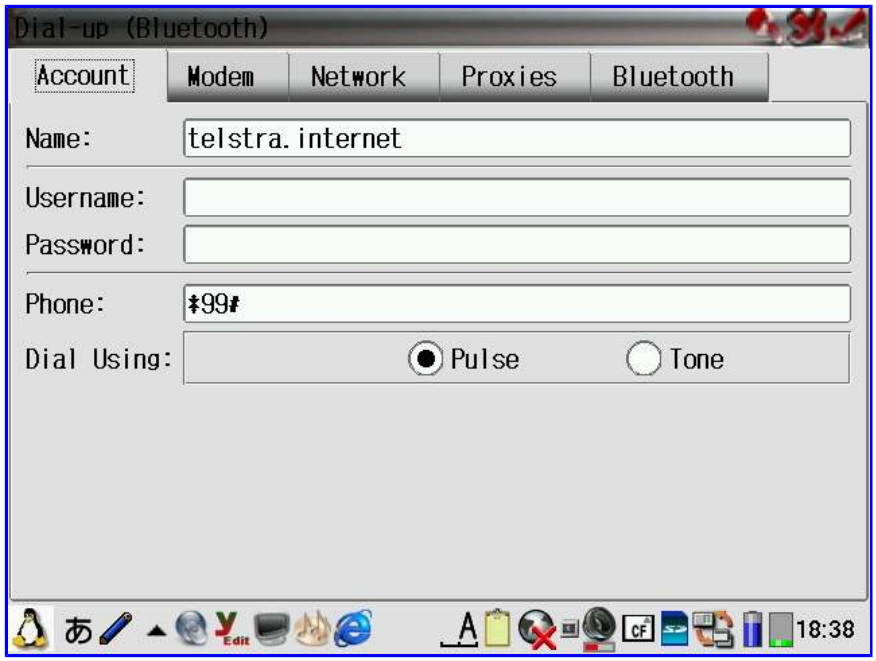
You can also just simply install irnet_2.4.20_arm.ipk which installs and configures irnet so that it will be available even after a reboot. It also provides an opie shell script to allow you to start and stop it from the Qtopia GUI or simply run **irnet start** to start it from the command line and **irnet stop** to stop it.

Using Bluetooth for networking:

If you have a CF bluetooth card or a bluetooth USB dongle, then you can set up a PAN (Personal Area Network) with other bluetooth enabled devices.



In the case you have a bluetooth enabled mobile phone with GPRS service, then you can even use bluetooth to connect to your phone using dialup networking (DUN) to use the phone's GPRS service.



However, the default Sharp ROM does not have bluetooth drivers or tools installed out of the box and you will need to setup and install those first before you can use bluetooth. See the bluetooth section for more details.

File Sharing and Services

Enabling Samba (over wireless or ethernet)

By default, a Samba service is already installed on the Zaurus. It is used when you synchronise your Zaurus with your PC while the USB cable is plugged in. You can also manually start and stop the Samba service and allow it to go over your wireless and ethernet network instead of just the USB cable. If you use the USB cable in advanced mode (with TCP/IP enabled), then you will be able to access all your devices (MicroDrive, SD card, CF card and USB disk) at the same time instead of being able to only chose one at a time in normal sync mode.

To allow Samba to be accessed via the WLAN (wlan0, wifi0) or LAN (eth0) interface, edit the following file: /usr/lib/samba/smb.conf

Find the following line: *interfaces = usbd0*

add your network interface after *usbd0* separated by a space like this: *interface = usbd0 wlan0 eth0*

You might also want to add a new entry: *hosts allow = 192.168.1.* (whatever the IP range of your network is from which you want to connect to your Zaurus, multiple entries are separated with a space)

```
[global]
workgroup = HOME
log file = /dev/null
hosts allow = 192.168.1. 192.168.129.
encrypt passwords = yes
coding system = utf8
client code page = 932
force create mode = 0755
strict sync = yes
sync always = yes
interfaces = usbd0 eth0 wlan0
#wins support = yes
bind interfaces only = yes

[system]
comment = System Folder
path = /root/samba
read only = no
browseable = no
guest ok = yes
force user = root

[home]
comment = for User Data
path = /home/samba
short preserve case = no
read only = no
guest ok = yes
force user = zaurus
```

And finally, you need to know how to start the Samba service:

```
# su
# /etc/rc.d/init.d/samba start
```

To stop the Samba service:

```
# su
# /etc/rc.d/init.d/samba stop
```

Alternatively, you can install sambacontroller [sambacontroller_0.1-0_arm.ipk] which gives you a GUI interface to do it. Remember to give it root access or else nothing will happen. Also you need to do the following to enable it to configure smb.conf (only needed for the C3000)

```
# su
# mkdir -p /home/root/usr/lib/samba
# cd /home/root/usr/lib/samba
# ln -s /usr/lib/samba/smb.conf smb.conf
```

Once you install SambaController, you can use it to start and stop samba, as well as modify smb.conf from the Configure tab.

You should also install smbpasswd [smbpasswd_0.1-1_arm.ipk] which will allow you to set the samba password.

Once you installed smbpasswd, you will need to do the following to get it working:

```
# su
# ln -s /usr/local/samba/lib/codepages /usr/lib/samba/codepages
# rm /usr/local/samba/lib/smb.conf
# ln -s /usr/local/samba/lib/smb.conf /usr/lib/samba/smb.conf
# ln -s /usr/local/samba/private/smbpasswd /etc/smbpasswd
```

Note: you might need to hack the Windows registry to enable plain text password to make Windows compatible with Samba (depends on the version of Windows you are running).

This configuration forces the samba user to be the zaurus. When giving access to the samba shares, all the access rights and file permission is that of the zaurus user. This will work fine on all FAT filesystems, however, if you change some of your partitions to ext2, then you might get some permission denied errors when trying to acces, create, modify, move or delete files. If that happens, make sure the file or directory in question has sufficient access rights for the zaurus user.

In addition, if you change /hdd3 to another filesystem other than fat, then samba will stop to work for /hdd3/Documents unless you change /etc/hotplug/usbdstorage.agent and change the mount options to the appropriate filesystem type, ie change fat to someting else like ext3.

Installing SSH

ssh client

If you only want to ssh to other machines from your Zaurus, install ssh client [openssh-client_3.6.1p1_arm.ipk].

sshd

If you want your Zaurus to be accessed via ssh as well, then you need to install the following:

- openssl - [openssl_0.9.7d_arm.ipk]
- openssh-server [openssh-server_3.6.1p1_arm.ipk]
- openssh-addon - [openssh-addon_3.6.1p1_arm.ipk]

Enabling telnet

telnet client

There already is a command line telnet client pre-installed on the Zaurus which you can use.

telnetd

In order to enable the telnet daemon within inetd, uncomment the telnet entry in /etc/inetd.conf and restart inetd.

```
# su
# vi /etc/inetd.conf

    telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd

# /etc/rc.d/init.d/inet restart
```

Make sure you use tcp wrapper as it is slightly more secure. Then enable tcp wrapper security by creating hosts.allow and hosts.deny as follows:

```
# su
# echo "ALL:ALL" > /etc/hosts.deny
# echo "in.telnetd: 192.168.129., 192.168.1." >> /etc/hosts.allow
```

Add any IP range you want to give access to in addition to the above.

Installing FTP

ftp client

A command line ftp client is already installed, however, there are much nicer ftp clients such as ncftp [ncftp_3.1.5-1_arm.ipk] and lftp [lftp_2.6.7-1_arm.ipk]. Alternatively, there are also GUI based ftp clients such as opie-ftp [opieftp_0.9.1-20020702_arm.ipk] and jftp [jftp_0.23.1_arm.ipk].

ftp server

If you want to serve as a ftp server then you need to install utftp [utftpd_0.2.4_arm.ipk] or troll-ftpd [troll-ftpd_1.28-cg2_arm.ipk]. Alternatively, you can ftp to port 4242 on the Zaurus which is a very basic ftp service.

Alternatively, you can also enable the ftp daemon within inetd. To do that, uncomment the ftp entry in /etc/inetd.conf and restart inetd.

```
# su
# vi /etc/inetd.conf

    ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a

# /etc/rc.d/init.d/inet restart
```

Make sure you use tcp wrapper as it is slightly more secure. Then enable tcp wrapper security by creating hosts.allow and hosts.deny as follows:

```
# su
# echo "ALL:ALL" > /etc/hosts.deny
# echo "in.ftpd: 192.168.129., 192.168.1." >> /etc/hosts.allow
```

Add any IP range you want to give access to in addition to the above.

Installing Web Server

Apache

There is a few things that need to be done before you can install Apache:

```
# su
# mkdir -p /hdd2/ramfs/www
# ln -sf /hdd2/ramfs /mnt/ramfs
# ln -s /hdd2/ramfs/www /usr/local/apache
```

Now we are ready to install apache [apache-1.3.27-php-4.2.3_0.1_arm.ipk]

When apache is installed you need to do the following:

```
# su
# ln -s /usr/local/apache/bin /usr/local/apache/src
```

(you can also fix apachectl to look at the right place instead of creating a link)

You can now start apache with the following command:

```
# su
# cd /usr/local/apache/bin
# ./apachectl start
```

You can stop apache with the following command:

```
# su
# cd /usr/local/apache/bin
# ./apachectl stop
```

I have also created a opie-sh script [apachegui_0.1_arm.ipk] which allows you to stop and start apache from the GUI. You will be required to configure sudo and add /usr/bin/apache to the allowed list for zaurus user (see sudo).

Alternatively, you can also install boa [boa_0.94.12_arm.ipk] which is a light-weight http daemon if you just want simple web server.

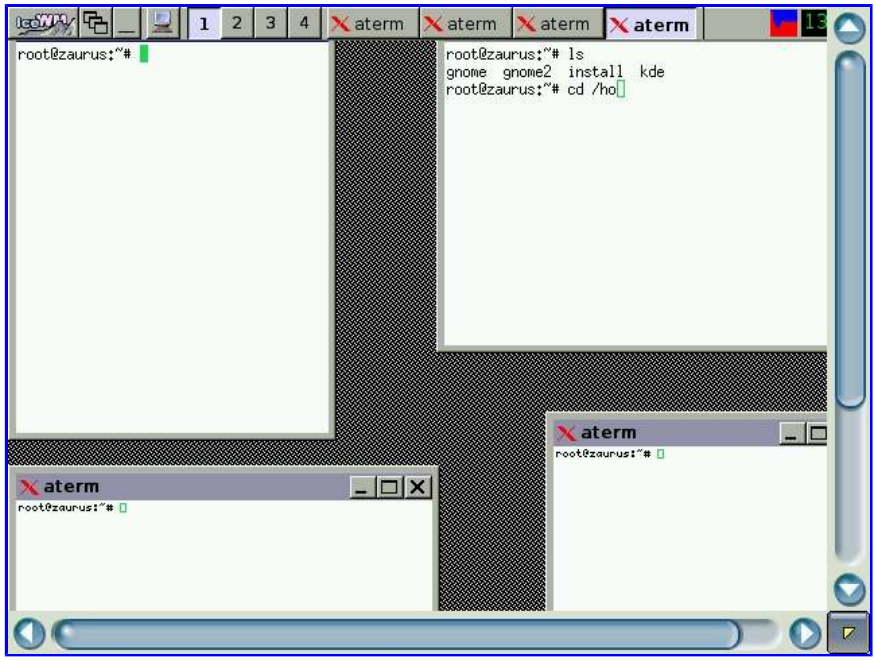
Browsers

There are several browsers available such as NetFront, Opera, Konqueror, Firefox, Minimo, Dillo, Links, ...

Installing VNC

VNC client

keypepple [opie-keypebble_1.0.0-1_arm.ipk] can be used as a vnc client to other machines or even the local one.



VNC server

fbvnc server for Qtopia

You can run a vncserver on your Zaurus to enable remote access to its desktop. However, since the available fbvnc server packages were build for other models, it does not work reliably on the C3000 and C3100.

I have build my own version of fbvncserver [fbvncserver-c3000_0.9.4-0.3_arm.ipk] which allows me to view the Zaurus desktop remotely using a vnc client such as tightvnc or using a web browser to connect to port 5800 on the Zaurus.

I also managed to get the mouse pointer working through the vnc server as well as the keyboard. However, some keys are still wrongly mapped. If you just want a read only vnc server without remote mouse and keyboard entry, then do the following to disable them:

```
# su
# rm /etc/rc.d/init.d/fbvncinput
# reboot
```

Alternatively, you could also just restart Qtopia instead of rebooting the Zaurus. To do that instead do the following:

```
# su
# /usr/local/bin/fbvncinput stop
# killall qpe
```

If you want to re-enable the keyboard and mouse, then just recreate the the symbolic link to fbvncinput and reboot your Zaurus:

```
# su
# ln -s /usr/local/bin/fbvncinput /etc/rc.d/init.d/fbvncinput
# reboot
```

Alternatively, you could also just restart Qtopia instead of rebooting the Zaurus. To do that instead do the following:

```
# su
# /usr/local/bin/fbvncinput start
```

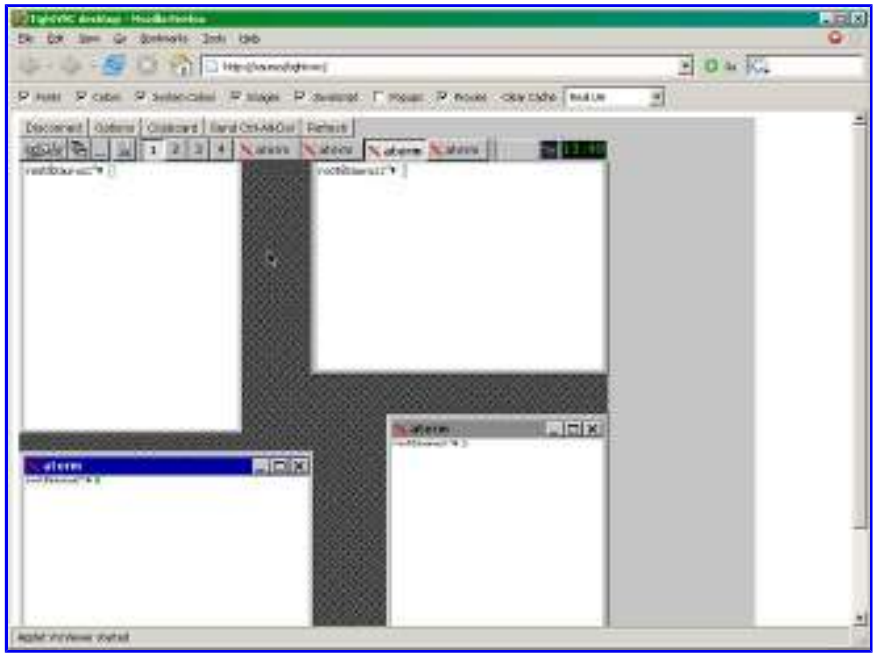
```
# killall qpe
```

Vncserver for Debian

Vncserver is part of the Debian PocketWorkstation bundle and is intended as a loopback service to connect to the Debian instance locally. However, there is nothing preventing it from being accessed remotely as well. Vncserver listens on port 5901 and can be easily accessed through keypebble and/or tightvnc [tightvnc-1.2.9_javabin.zip]. tightvnc is a java application and can be installed on any machine that runs java. It can also be installed under a webserver such as Apache. Simply make a directory such as **vnc** under the document root and copy index.html and VncViewer.jar into there.

```
# mkdir -p /mnt/ramfs/www/htdocs/vnc
# cp index.html /mnt/ramfs/www/htdocs/vnc
# cp VncViewer.jar /mnt/ramfs/www/htdocs/vnc
```

Then just point your browser at it: http://zaurus/vnc



Mounting Filesystems

Mounting Samba shares

Install the following packages:

- libncurses - [libncurses_5.0_arm.ipk]
- smbmount - [smbmount_0.1_arm.ipk]
- smbmounter - [smbmounter_0.1-2_arm.ipk]

To mount a drive type:

```
# smbmount //hostname/share /mnt/smb -o username=user
```

To unmount type

```
# umount /mnt/smb
```

Or use the smbmounter GUI interface. Make sure you give it root access (see Run as root section). The NETBIOS name needs to be an IP address or if you provide a hostname, make sure in your /etc/hosts file on your Zaurus you have given the hostname an alias in all uppercase letters. For example if you have a hostname mylaptop mapped to an IP of 192.168.1.10 then you will need the following entry:

```
192.168.1.10 mylaptop MYLAPTOP
```

Once you have created an entry with a valid share name and username/password, then you can mount and unmount the share by clicking on the appropriate buttons. If the share is valid and your username/password is correct, then the smb share will be mounted under /hdd3/Documents/NetworkFolders/*hostname*/*sharename*

Mounting USB drives

To mount a drive type:

```
# mount /dev/sda1 /mnt/usbstorage
```

To unmount type:

```
# umount /mnt/usbstorage
```

You can also create the following simple script to automatically mount USB drives or use the more advanced usb-storage that I've written. Here is the simple version:

```
# su
# vi /etc/hotplug/usb/usb-storage
```

```
#!/bin/sh
. /etc/hotplug/hotplug.functions
if [ ! -L /var/run/usb/%proc%bus%usb%* ]; then

        mesg Try to Mount
        mount /mnt/usbstorage
        if [ $? = 0 ]; then
                ln -s /mnt/usbstorage /home/samba/USB_Storage
                ln -s /etc/hotplug/usb/usb-storage.off $REMOVER
                mesg make REMOVER in $REMOVER
        fi

fi
```

```
# chmod 755 /etc/hotplug/usb/usb-storage
# vi /etc/hotplug/usb/usb-storage.off
```

```
#!/bin/sh
. /etc/hotplug/hotplug.functions
mesg Removing /mnt/usbstorage
rm /home/samba/USB_Storage
umount /mnt/usbstorage
```

```
# chmod 755 /etc/hotplug/usb/usb-storage.off
# echo "/dev/sda1 /mnt/usbstorage vfat noauto,umask=000, noatime,iocharset=utf8,codepage=932
0 0" >> /etc/fstab
```

Hint: Once a USB disk is mounted, it will appear in the **Files tab** as well. In addition, creating a link to the mounted drive from /home/samba will allow it to be shared as well through Samba. In general, most devices such as memory sticks, cameras, mp3 players and usb harddrives have their disk partitioned as

a primary partition and can be found at /dev/sda1. However, if you have partitioned your disk as an extended partition, then it most likely would be /dev/sda5. Do a **fdisk -l /dev/sda** as root to check and change the above accordingly.

In addition, if you are using a USB Hub, then you will be able to attach and mount multiple devices, usually up to four disks. In such a case, the devices will be /dev/sda, /dev/sdb, /dev/sdc and /dev/sdd. I have created a more advanced script which can automatically mount up to four usb disks and also automount disks partitioned as primary or extended partitions. Place this usb-storage into /etc/hotplug/usb and remove usb-storage.off if there already is one (it will generate a new one). The script will also create mount points under /mnt and update /etc/fstab as required. Alternatively, you can simply install [automounter-c3000_0.5.0_arm.ipk].

In addition to external harddisk enclosures with USB interfaces and memory sticks, most MP3 players, cameras and some mobile phones also have an internal storage that can be mounted on the Zaurus if they have a USB interface. Most of those devices will be recognised by the Zaurus as a Mass Storage device by default, however, some newer devices are not on the Zaurus' device list and you will need to update it to let the Zaurus know about the new device (see SonyEricsson section for an example).

The automounter script only automatically mounts the first mountable partition. If you have multiple partitions, then you will need to manually mount the remaining partitions or modify the automounter script to also mount the remaining partitions.

The C3000 and C3100 can also read NTFS formatted USB disks. You will need to copy ntfs.o to /lib/modules/2.4.20/kernel/fs/ntfsfs/ or install [ntfs-zaurus_2.4.20_arm.ipk].

A useful applet that I wrote [qtopia-usbapplet_1.0.3_arm.ipk] can be used to unmount the USB disks prior to unplugging them without needing to go to the command line.

Mounting SD and CF cards

The SD and CF cards are automatically mounted when they are inserted. In addition to mounting and unmounting them, additional hooks and controls can be added to the scripts such as invoking /etc/rc.d/init.d/mntloop (automounter) to check for the presence of a swapfile and enabling it when the card is mounted and disabling it when the card is unmounted. The SD card is auto mounted by the /etc/sdcard/sd_mem_ctrl script and this is where the control hooks need to be added. The CF card is auto mounted by /etc/pcmcia/ide.

Furthermore, by default, the SD card script only attempts to mount the first partition. If you have multiple partitions, then you need to enhance the above mentioned script to automatically mount additional partitions on the CF or SD cards.

Remounting Filesystem as Read/Write or Read Only

Some partitions/file systems such as / cannot be unmounted. Some are also mounted as read-only. In order to modify the files on those file systems, you need to remount them as read/write and then remount them back to read-only after you have done what you wanted.

To remount as read/write:

```
# mount -o rw,remount /
```

To remount as read only:

```
# mount -o ro,remount /
```

Enabling Swap

The following demonstrates how to configure a 128MB swap file on the C3000's harddisk. Normally a swap file is not required unless you are running many X/Qt applications and doing onboard development. Most users with X/Qt will probably only require 32MB - 64MB swapfile.

```
# su
# dd if=/dev/zero of=/hdd3/swapfile bs=1048576 count=128
```

```
# mkswap /hdd3/swapfile
# swapon /hdd3/swapfile
# echo "/hdd3/swapfile swap defaults 0 0" >> /etc/fstab
```

To enable swap:

```
# su
# swapon /hdd3/swapfile
```

To disable swap:

```
# su
# swapoff /hdd3/swapfile
```

To check swap status:

```
# cat /proc/swaps
```

In addition, you can install [qtopia-memoryapplet_1.0.4_arm.ipk] which allows you to monitor both your physical memory as well as your swap. It even can manage the swapfile creation for you. This version can create a swapfile with a max. of 512MB. It also gives you the option to create the swap on the internal HDD (/hdd3), CF card and SD card. The created swap file is also called **swapfile** rather than **.swapfile**.

Increasing tmp

By default /tmp is mounted from /dev/shm as a 1MB tmpfs. This 1MB is taken from the 64MB of RAM and for most applications, 1MB of tmp is sufficient. However, some applications such as kismet or qpdf2 may require a bit more than just 1MB. Opening some large pdf files for example might be very slow or does not work at all because there is insufficient space in /tmp. Some applications can utilise /home/root/tmp instead, but not all can.

You can increase the amount of memory allocated to /tmp. But remember that memory for /tmp is taken from RAM, so increasing /tmp will decrease the amount of physical available RAM.

To do that, you need to edit /etc/fstab and /root/etc/rc.d/rc.rofilesys and replace 1m with for example 2m to increase the size of /tmp to 2MB. Remember to remount / to rw before editing rc.rofilesys and remount it back to ro once done. Finally, you need to reboot the Zaurus in order for the change to take effect.

Using loopback filesystem

You can use a loopback filesystem which is a mounted filesystem image to do various things such as compressing files or overlaying a filesystem with another format.

The cram filesystem is a read only compressed filesystem format. The following demonstrates how to configure cramfs to preserve some space. Only convert directories to cramfs if you are sure those directories are read-only, ie you are never going to change or add any files there. You will need either [cramfs-1.1_arm.bin.tar.gz] or [mkcramfs.tar.gz], or install gcc which also includes it. Here is an example for compressing and mounting the jre directory:

```
# su
# mkcramfs /usr/lib/jdk1.3 /hdd3/jre13.cramfs
# rm -r /usr/lib/jdk1.3/*
# mount -t cramfs -o loop /hdd3/jre13.cramfs /usr/lib/jdk1.3
```

So far I have crammed the following:

- /usr/lib/jdk1.3
- /usr/lib/firefox0.9gtk

- /usr/lib/thunderbird-0.6

Note that **mkcramfs** stores the whole image in memory before writing it to disk, so make sure you have a sufficiently sized swap file enabled before running **mkcramfs**.

Also, there are only 2 loop devices by default on the C3000 and C3100, but you can create more loop devices. You will need to recreate them each time you reboot, so it would be better to automate it in a start script which you can use to automatically mount the cram archives as well. Here is how you create /dev/loop2 to /dev/loop6

```
# for i in 2 3 4 5 6
> do
> mknod /dev/loop$i b 7 $i
> done
```

You might also want to add an entry into fstab so the cramfs archives can be automatically mounted. Copy mntloop to /etc/rc.d/init.d and link it to rc5.d and rc6.d

```
# su
# cp /home/zaurus/Documents/custom/mntloop /etc/rc.d/init.d
# ln -s /etc/rc.d/init.d/mntloop /etc/rc.d/rc5.d/S50mnt
# ln -s /etc/rc.d/init.d/mntloop /etc/rc.d/rc6.d/K50mnt
# echo "/hdd3/jre13.cramfs /usr/lib/jdk1.3 cramfs loop,ro 0 0" >> /etc/fstab
```

mntloop will create new loop devices and check /etc/fstab at bootup time and mount any valid entry for cramfs it finds. The Linux boot process will mount entries in /etc/fstab by default, but it can only mount 2 loop partitions because there are only that many default loop devices, so any additional entries in /etc/fstab will fail to mount. *mntloop* will mount any additional entries it finds after creating the extra loop devices. At system shutdown or reboot, it will cleanup and unmount anything mounted as a loop device.

The automounter package [automounter-c3000_0.5.0_arm.ipk] will install mntloop and also install the usb automounter.

Similar to the cram filesystem is the squash filesystem which is appendable, ie you can add files to it. In order to use the squash filesystem, you will need to install the squashfs module [kern-mod-squashfs_c3000-2.1-2_arm.ipk].

Alternatively, if you don't need a compressed filesystem but want a read and write access instead, then you can create an ext2 or ext3 formatted loopback filesystem. You would need to pre-allocate a chunk of space for it and the files won't be compressed at all.

For example, to create a 128MB loopback filesystem on /hdd3 called expansion.ext2 do the following:

```
# su
# dd if=/dev/zero of=/hdd3/expansion.ext2 bs=1MB count=128
# echo y|/sbin/mke2fs /hdd3/expansion.ext2
# mkdir -p /home/expansion
# mount -o loop -t ext2 /hdd3/expansion.ext2 /home/expansion
# echo "/hdd3/expansion.ext2 /home/expansion ext2 loop,rw,noatime 0 0" >> /etc/fstab
```

Alternatively, if you rather create an ext3 filesystem instead of an ext2, do the following:

```
# su
# dd if=/dev/zero of=/hdd3/expansion.ext3 bs=1MB count=128
# echo y|/sbin/mke2fs -j /hdd3/expansion.ext3
# mkdir -p /home/expansion
# mount -o loop -t ext3 /hdd3/expansion.ext3 /home/expansion
# echo "/hdd3/expansion.ext3 /home/expansion ext3 loop,rw,noatime 0 0" >> /etc/fstab
```

Now you could move all the files from /usr/local to the newly mounted loopback filesystem and then relink /usr/local to the loopback filesystem. This way, we moved some files off to hdd3 and extended the

/usr/local directory to have a higher limit of 128MB. Similarly you could do something similar with /opt/QtPalmtop/share and many other directories as well. However, doing this will impact on performance a little bit. The overhead caused by the loopback filesystem and the speed of the MicroDrive are factors to consider and thus, choose files and directory that are not frequently used to be moved to the loopback filesystem.

On C3000:

```
# mkdir -p /home/expansion/usr/local
# cd /usr/local
# tar cvf - * | tar xvf - -C /home/expansion/usr/local
# mv /hdd2/usr/local /hdd2/usr/local.bak
# ln -s /home/expansion/usr/local /hdd2/usr/local
```

Once you have tested that everything still works, you can remove /hdd2/usr/local.bak

On C3100:

```
# mkdir -p /home/expansion/usr/local
# cd /usr/local
# tar cvf - * | tar xvf - -C /home/expansion/usr/local
# mv /home/root/usr/local /home/root/usr/local.bak
# ln -s /home/expansion/usr/local /home/root/usr/local
```

Once you have tested that everything still works, you can remove /home/root/usr/local.bak

Make sure you install the automounter package [automounter-c3000_0.5.0_arm.ipk] or else your loopback filesystem might not get automatically mounted after a reboot.

Alternatively, instead of moving files manually after they have been installed, you can also just install files to the loopback filesystem directly. The default installer won't let you do it, but my **xipk** install script (which is part of my ipktools package) does allow it.

By default xipk installs to /hdd3/programs, so you should mount your loopback filesystem as /hdd3/programs or change /etc/xipk.conf to contain the mount point of your loopback filesystem. Then you can do the following to install applications to hdd3:

```
# xipk ipkfile
```

Enable large SD cards

The default SD/MMC driver only supports SD and MMC cards of sizes up to 1GB. Using the updated driver which is taken from the C3200, it is possible to use larger SD cards. 2GB and 4GB SD cards are recognised and can be used once this updated driver is installed.

However, some applications have 1GB or 2GB (upper limit for FAT16 partitions and maximum filesize on FAT16) hardcoded as the upper limit and thus will miscalculate the amount of free space on the larger SD card. Also, this driver is not loaded during emergency boot or the NAND loader so you cannot use the larger SD card to flash your Zaurus or do NAND backup/restore.

Also, for 4GB SD cards, be very careful when ejecting the card. If you eject it while it is still mounted or while it is being written to, then you might corrupt the integrity of the device and might not be able to use it anymore. Since it is larger, it needs more time to flush the buffers and thus this problem occurs less on smaller SD cards. If this occurs, however, even a fdisk or reformatting of the SD card won't work (the smaller SD cards can be reformatted in some digital cameras but not many cameras can recognise the larger SD cards either so you cannot use them to reformat a broken SD card). To prevent this from happening in which case you need to claim a warranty replacement so make sure you got proper warranty when you purchase large SD cards, you can change the SD mount options to mount the SD card with the **sync** option and also increase the wait delay from 1 second to 3 or 5 seconds during the SD unmount process.

To make the SD card mount with the sync option, modify */etc/sdcard/sd_mem_ctrl change the following*

from

```
FATOPTS="-o noatime,umask=000,iocharset=utf8"
OPTS="-o noatime"
```

to the following

```
FATOPTS="-o noatime,umask=000,iocharset=utf8,sync"
OPTS="-o noatime,sync"
```

To increase the wait delay during the SD unmount, modify */etc/sdcontrol* and change the following from

```
sleep 1
```

to the following

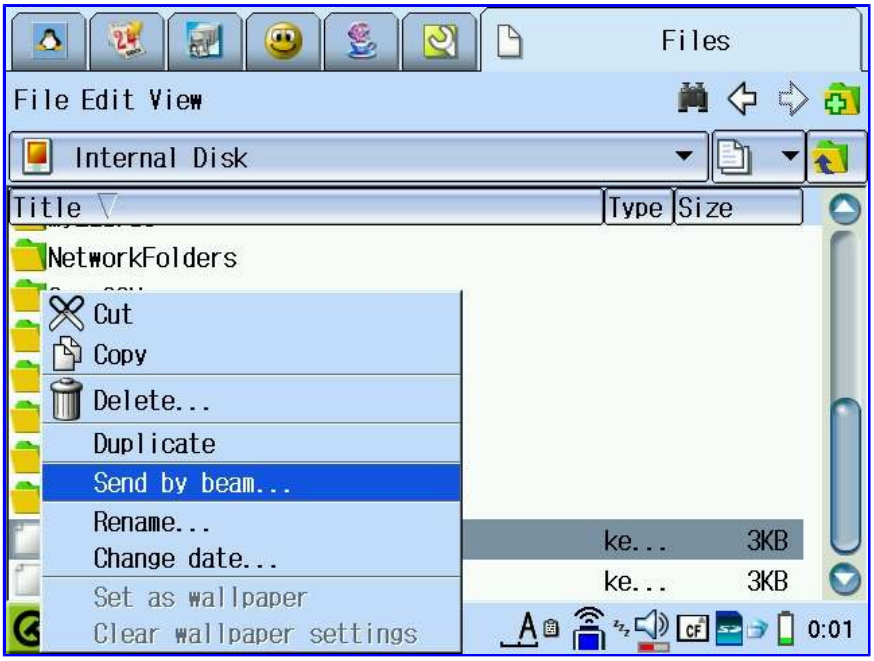
```
sleep 3
```

Enabling IrDa

The Zaurus has an IrDa port build-in, but for security and power saving, it is disabled by default. You can temporarily enable it to receive files. Use the IR-Receive tool under the Settings tab and enable it to receive files. You should disable it once you finished receiving files.

Beam Receive
OK
Beam receive mode
Manual
Automatic
Since this device regularly looks for sending device in "Automatic" mode, it consumes the battery.
"Manual" setting is recommended if you don't use beam transfer for a long time.
Manual beam receive
Please press "Start" button to receive data.
When it is done, press "Stop" button.
Start
9:07

You can also send files via IrDa. For that, select the file you want to beam from the Files Tab and hold the stylus on it for a few seconds and select *Send by beam...*. I have tested this feature and it works fine to send and receive files to my Laptop (Toshiba Libretto 50CT) and Mobile phone (Sony Ericsson K750i).



Enabling Bluetooth

The stock Sharp ROM does not come with bluetooth enabled. You can add bluetooth capability to the Zaurus by either using a CF Bluetooth card or a USB Bluetooth dongle. I've tested this using a Socket Bluetooth CF card which I think is great because it is exactly the same size as a CF memory card. It has a CF I form factor and no bits sticking out when inserted into the CF slot on the Zaurus. I have also tested this with a tiny WIDCOMM compatible USB Bluetooth dongle.

In order to enable and use Bluetooth, you need to install a bluetooth stack such as bluez which includes the required kernel modules as well as needed command line tools. You can also add some plugins to the graphical Network config tool to enhance it to handle bluetooth connection types. There are currently two such plugins available, one for PAN (Personal Area Network) and one for DUN (dial up bluetooth). Once they are installed, you will see additional options in your Network config tool.

Here are the files you need:

- bluez-zaurus_2.12_2.4.20_alpha4_arm.ipk
- bluepin_0.0.1-1_arm.ipk
- susp_resume_bluez_0.9.3_arm.ipk
- qtopia-bluetoothnetworkapplet_1.0.1_arm.ipk
- qtopia-pannetworkapplet_1.0.1_arm.ipk

The bluez package is essential while the others are optioanl. If you find newer updated versions of those, then use them instead. Once you have installed the above packages, you can begin to setup and configure your bluetooth stack. I have also created a single package [bluetooth-support_1.23_arm.ipk] which contains the above as well as additional obex packages. There is also a bluetooth lite package which only contains the GUI scripts. The qshdlg package needs to be installed to use the bluetooth GUI.

The first thing you need to do is check your config and make sure your bluetooth stack has initialised successfully.

```
# su
# /etc/rc.d/init.d/bluetooth restart
# hciconfig
```

You will see something like this:

```
hci0: Type: UART
        BD Address: xx:xx:xx:xx:xx:xx ACL MTU: 672:10 SCO MTU: 64:0
```

```
UP RUNNING PSCAN ISCAN
RX bytes:250 acl:0 sco:0 events:12 errors:0
TX bytes:446 acl:0 sco:0 commands:12 errors:0
```

Then you need to search for bluetooth enabled devices you want to use and do the pairing.

```
# hcitool scan
```

Note that unless the other bluetooth devices are configured to be visible, you won't be able to find them. Write down their mac addresses. (xx:xx:xx:xx:xx:xx is the format for the mac addresses and each device will have a unique address), eg:

```
xx:xx:xx:xx:xx:xx K750i
xx:xx:xx:xx:xx:xx OQOU1
xx:xx:xx:xx:xx:xx C3000
```

To find out the capabilities of those devices do the following with their corresponding mac address:

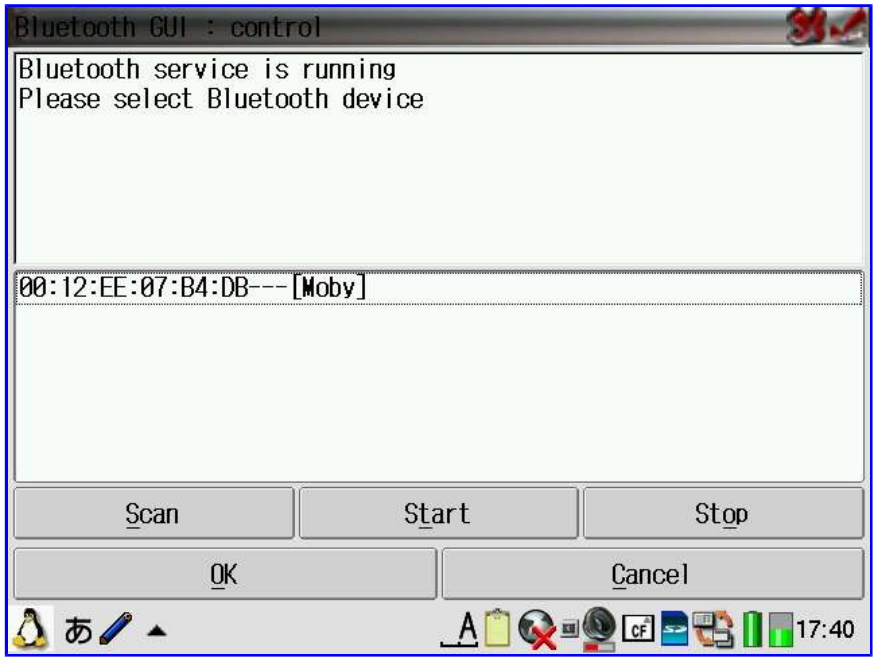
```
# sdptool browse xx:xx:xx:xx:xx:xx
```

You can initiate the pairing from each of those devices and make sure the PIN matches. The Zaurus stores its PIN in /etc/bluetooth/getpin and /etc/bluetooth/pin

/etc/bluetooth/pin contains the pin number that you enter in any other Bluetooth device that pairs with the Zaurus.

/etc/bluetooth/givepin contains the pin number that your Zaurus will automatically give to another Bluetooth device if pairing from the Zaurus. givepin is a script which must print out a string of the format "PIN:1234" where 1234 is the pin number, so you only change that part of the script to change to pin number.

Once the devices are paired with your Zaurus and you have determined what capabilities they have, you can setup and configure your Zaurus to use them.



The bluetooth qshdlg GUI or the network config tools plugins can be used to configure the following.

Setting up a bluetooth PAN (Personal Area Network)

A PAN allows you to access services on the other bluetooth enabled devices like on a small network with TCP/IP connectivity. This requires that the other devices, usually bluetooth enabled computers, provide services such as Samba, FTP, HTTP, Telnet, SSH/SCP, etc.

On your Zaurus do the following:

```
# su
# modprobe bnep
# pand --role PANU --service NAP --connect xx:xx:xx:xx:xx:xx --nodetach
```

(where xx:xx:xx:xx:xx:xx is the mac address of your Desktop PC or Laptop)

This will give your Zaurus an additional network interface called bnep0 if a successful connection was established.

```
# ifconfig -a
```

You can then configure your PAN network like this:

```
# ifconfig bnep0 192.168.12.201
# route add default gw 192.168.12.10
```

To check the connection ping your desktop or laptop:

```
# ping 192.168.12.10
```

This assumes that you have a 192.168.12.0 private network and your PC or Laptop is 192.168.12.10

To terminate the connection just type:

```
# pand -K
```

Alternatively, you could also use the PAN applet in the network config tool to do the same thing or use my bluetooth-gui script.

Setting up a bluetooth Dialup connection

You can also use bluetooth to behave like a serial communication device. You can connect to other devices such as computers via a serial line similar to an IrDA connection but with a wider range and greater speed, or connect to a modem on a computer or mobile phone to dialup internet services. Using the serial communication feature, you can emulate PPP and/or use OBEX for file transfer (the IrDA beaming feature).

To setup bluetooth dialup service, for example, you can do the following to dial a GPRS on a mobile phone. First create a config file under /etc/ppp/peers as follows:

```
/dev/rfcomm0
115200
connect '/usr/sbin/chat -s -v -t 60 ABORT "NO CARRIER" ABORT "NO DIALTONE" ABORT
"BUSY" "" "ATZ" OK "ATDP*99#" CONNECT'
crtscts
noipdefault
modem
usepeerdns
defaultroute
connect-delay 5000
remotename DUN1138428518
```

Make sure the remotename matches the id in
/home/zaurus/Applications/Network/modules/Bluetooth.conf

For the Telstra GPRS no username and password was required and the network settings were DHCP. The only thing needed to be provided was the GPRS profile name which was telstra. The default dial string of atz was sufficient with *99# as the number which tells the phone to use its local profile. Additionally, *99***telstra.internet# could had been specified as well to choose a specific profile during dialing.

Then do the following:

```
# su
# pppd file /etc/ppp/peers/yourconfig
```

Alternatively, you could also use the Bluetooth (DUN) applet in the network config tool to do the same thing.

Setting up beaming with bluetooth via OBEX

You can use OBEX to push and receive files similar to IrDa but with bluetooth instead.

To transfer files towards the device, you have to install the package obexftp [obextftp_0.10.7_arm.ipk]. To transfer a file, simply execute:

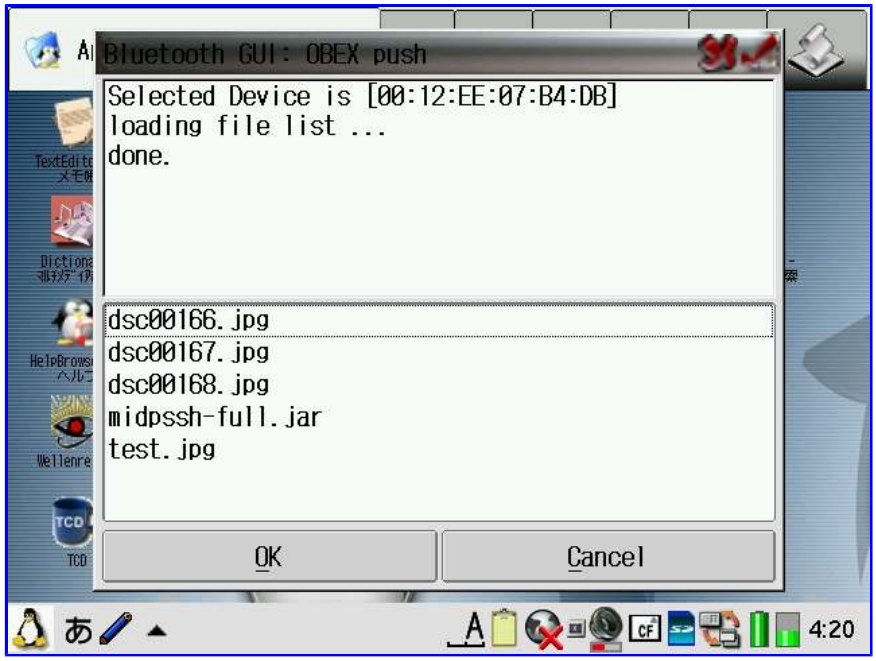
```
# obexftp -b 00:00:00:00:00:00 -p file.ext
```

The parameter -b instructs the program to use Bluetooth. If you omit the MAC address, obexftp will scan for surrounding Bluetooth devices and select the transfer partner automatically provided your device is visible and paired.

Alternatively, you can also use obextool:

```
# obextool push file.ext xx:xx:xx:xx:xx:xx 6
```

The bluetooth GUI allows you to convenniently select files to be beamed over.



To receive a file on your Zaurus via the Bluetooth interface, an OBEX server [obexserver_1.0_arm.ipk] has to be installed and running. The service "OBEX PUSH" also has to be registered to the SDP daemon:

```
# /usr/sbin/opd --mode OBEX+BIP --channel 4 --sdp --daemonize --path
/home/zaurus/Documents/Obex_Inbox
```

Securing Bluetooth

You can add some rudimentary security to bluetooth by editing /etc/bluetooth/hcid.conf:

- **Turn on encryption**
  remove the # in front of 'encrypt enable;'
- **Hide your Zaurus so it cannot be discovered**
  change 'iscan enable;' to 'iscan disable;'
- **Disable connection to your Zaurus**
  change 'pscan enable;' to 'pscan disable;'

You need to restart the bluetooth stack in order for the changes to take effect:

```
# su
# /etc/rc.d/init.d/bluetooth restart
```

Enabling Mouse

Yes, you can plug your USB mouse into the Zaurus but it won't work until you do the following:

```
# su
# cd /home/QtPalmtop/plugins/applets
# cp /home/zaurus/Documents/custom/libusbmouseapplet.so.1.0 .
# ln -s libusbmouseapplet.so.1.0 libusbmouseapplet.so.1.0.0
# ln -s libusbmouseapplet.so.1.0 libusbmouseapplet.so.1
# ln -s libusbmouseapplet.so.1.0 libusbmouseapplet.so
```

This custom driver package [zmouse_0.1_arm.ipk] should do the same thing.

Restart the Zaurus (or just restart Qtopia)

Alternatively, you can also install inputhelper [inputhelper_1.0.1-1_arm.ipk] which will give you mouse support as well as macro recording for your keyboard.

Enabling USB Keyboard

Yes, you can plug your USB keyboard into the Zaurus and it will work, but some of the keys will be mis-matched unless you are using a Japanese keyboard because the Zaurus' default keyboard layout is the Japanese keyboard. You need to remap the keyboard so the keystrokes matches that of the USB keyboard, but this of course will mess up the build-in keyboard on the Zaurus. You will need to switch between native keyboard and external keyboard mappings. You cannot have both keyboards mapped correctly at the same time unless your USB keyboard has the same layout as the Japanese keyboard or you hack the usbkbd kernel module.

If you want to remap the keyboard so that it is correctly mapped for an external USB keyboard, then you need to install usbkbd-en [usbkbd-en_2.4.20_arm.ipk] which once installed will automatically switch your keyboard layout depending whether your USB keyboard is plugged in or not. So with this replacement driver installed and its associated scripts, if you plug in a USB keyboard, your keymap will be automatically remapped for a US/AU QWERTY keyboard. When you unplug the keyboard, then the keymap is reverted back to the previous keymap settings.

To manually control the keyboard remapping run the following to enable the USB keymap:

```
# usbkey enable
```

To disable the USB keymap and revert back to the original map, type the following:

```
# usbkey disable
```

In addition, you can also modify the keytable so that on a USB keyboard, the numeric keys on the keypad behave just like you press Shift and a number key on the top row. To get this behaviour, you need to install dualkbd_2.4.20_arm.ipk.

Enabling Joypad

You may be able to use your USB Joypad. Install the joyenabler script [joyenabler_1.3_arm.ipk] and plug in your joypad. If you are lucky, it will work and you can use **jstest** to check whether it works correctly. Now you just need to find some games that support joypad controls. You may need an additional joypad driver for your particular joypad if it does not work.

Enabling WebCam

You can use some USB webcams. You will need Video4Linux support for the kernel (videodev.o) and an appropriate video driver for the webcam. I currently have compiled drivers that work with four of my six

webcams. You will also need some framegrabber software to capture video frames. I am currently working on some.

Enabling CD-ROM

USB CD-ROM drives are also supported similarly to USB harddisks. You will need to install some applications to rip and burn CD-ROMs [cdrtools-2.01.tar.gz]. A USB Mini-CD drive or a slim combo CDRW/DVD drive would be a great companion for the Zaurus.

The CD-ROM will be detected as /dev/scd0 and can be mounted as follows:

```
# su
# mkdir -p /mnt/cdrom
# mount /dev/scd0 /mnt/cdrom
```

In addition, the following will be useful:

```
# su
# mknod /dev/pg0 b 11 0
# mknod /dev/sg0 b 11 0
# ln /dev/scd0 /dev/cdrom
# ln /dev/scd0 /dev/vcd
# ln /dev/scd0 /dev/dvd
echo "/dev/cdrom /mnt/cdrom auto ro 0 0" >> /etc/fstab
```

In order to copy VCDs or DVDs from the CD to your disk, you need to install mplayer [mplayer-bvdd_1.1.5-1_arm.ipk] and mencoder [mencoder_1.1.0-1_arm.ipk]. A straight file copy won't work.

To copy the VCD video file (.dat file) do the following:

```
# mencoder vcd//:2 -oac lavc -ovc lavc -o filename.avi
```

To extract the vob files from a DVD do the following:

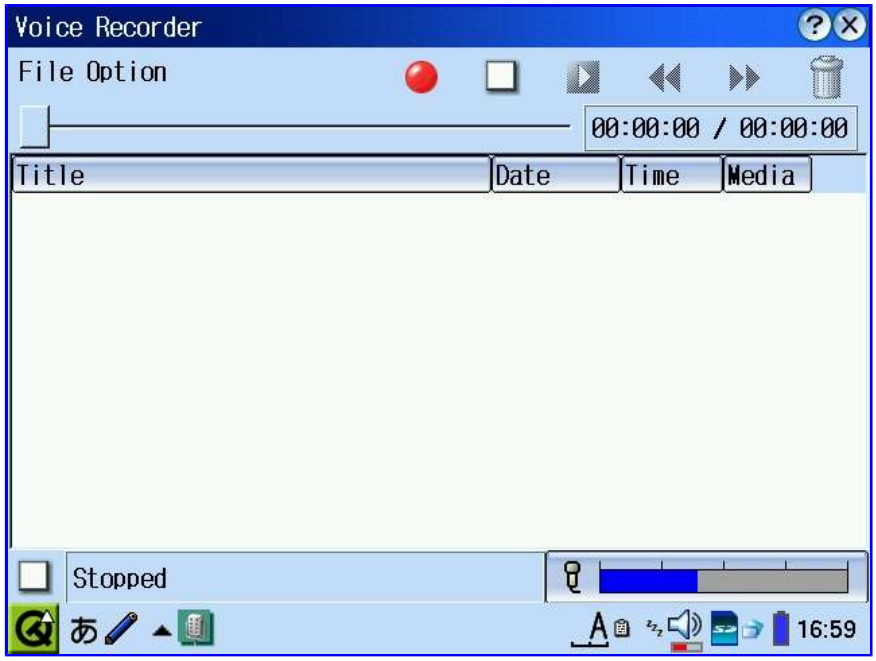
```
# mencoder dvd://# -ovc lavc -lavcopts vcodec=msmpeg4:vpass=1 -oac mp3lame -lameopts vbr=3
-o movie.avi
# mencoder dvd://# -ovc lavc -lavcopts vcodec=msmpeg4:vpass=2 -oac mp3lame -lameopts vbr=3
-o movie.avi
```

See Video Conversion section on how to re-encode and compress the videos for optimal viewing on the Zaurus.

Enabling Microphone

You can plug any 3.5mm microphone into the Zaurus. Since there is only one input connector, you will need to use a splitter if you want to plug in both a microphone and earphones at the same time, or use a combined headset with microphone such as the one for mobile phones and use a 2.5mm to 3.5mm adaptor.

There is nothing special you need to do in order to enable the microphone except install some recording software. There is an application called qpe-voicerec [qpe-voicerec_1.5.0-7_arm.ipk] that comes on the CD-ROM accompanying the Zaurus. You can use that to record some voice.



Alternatively, you can also use a console application called spxrec [spxrec_0.0.1_arm.ipk] to record lengthy voice sessions and then use shine [shine_0.01_arm.ipk] to encode to resulting sound file.

Connecting to VGA Monitor via USB

It is possible to connect the Zaurus to a VGA monitor or projector using a USB VGA dongle. The Kairen VGA adaptor is a USB VGA 2.0 adaptor that allows you to do slide show presentations using your Zaurus. You will need to use a USB hub to provide power as well as a USB host cable.

The Kairen USB2VGA dongle is detected as a sisusb vga device: USB2VGA dongle. It is allocated 8 output buffers with 8MB SDR SDRAM with a bus width of 32 by the driver.

The package vga-presentation_1.0.1_arm.ipk provides the driver and application for the USB dongle. I have only tested this specific model, but any USB dongle with the same SiS chipset (SiS 315E) should work also.

The VGA presentation application allows you to use the dongle for presentation. The application is limited to doing slideshows only. It slowly streams the data to the USB device which then buffers it and displays it to the monitor. I could even switch applications on the Zaurus while the presentation application continued to send data to the USB device and it renders it to the monitor. However, the VGA application crashes after a while so it needs a bit more work to iron out the bugs and fix the instability.

The image on the monitor remains even when rebooting the Zaurus so I guess the last image displayed is buffered by the USB dongle and the monitor is getting the cached data from it.

This proves that it is technically possible to have VGA out for the Zaurus via USB, however, the current state of the driver and application is very limited and needs more work. It is also very slow due to the slow USB bus speed implemented by the Zaurus.

I then plugged the USB dongle into one of my PCs and used it to mirror and split the display on a Win2000 system without any problems. The speed was ok there because it was using USB 2.0 High Speed instead of USB 2.0 Full Speed that the Zaurus uses.

At this stage, the USB dongle does not perform better than displaying via a VNC server on the Zaurus and a VNC client on a PC to display the Zaurus desktop on a larger monitor. This works fine if you can

network your Zaurus and the PC, but can't be done directly to a projector so this is where the USB dongle could be useful. With a USB dongle you do not need a PC or a laptop to act as the bridge device since the Zaurus can be directly connected to a projector or monitor.

For more info, visit the way-nifty blog where the images for the USB to VGA dongle and Presenter screenshot have been borrowed from.

Similarly, there are CF cards with VGA out capabilities similar to the USB dongle, however, none of those cards are manufactured anymore so you would need to get them second hand if you can find them. But even if you find those CF cards, you still need to compile drivers and applications for them if you manage to get the driver source code so they work on the C3x00, ie 2.4.20 kernel and glibc 2.2.2.

Connecting to Mobile Phone (SonyEriccson K750i)

The Sony Ericsson K750i is a mobile phone with many features. It has a 2 mega-pixel camera, and a slot for a Memory Stick Pro Duo card. I have upgraded mine to 1 GB. It also has Infrared, Bluetooth and comes with a USB interface which can be used for file transfer with the Zaurus. The phone also has a GSM modem via a serial line as well.

Using the USB cable to access the Memory Stick

The memory stick can be accessed as a mass storage device when the mobile is connected to the Zaurus via a USB cable. However, the Zaurus does not recognise the manufacturer id as a mass storage device by default. To fix that, you need the append the following to /etc/hotplug/usb.usermap:

```
usb-storage 0x000f 0x0fce 0xd016 0x0000 0x0000 0x00 0x00 0x00 0x08 0x06 0x50
0x00000000
usb-storage 0x0380 0x0fce 0xd016 0x0000 0x0000 0x00 0x00 0x00 0x08 0x06 0x50
0x00000000
```

This should make the Zaurus detect the K750i's Memory Stick as a Mass Storage device the next time it is plugged in. However, if it does not, then try appending the above into /lib/modules/2.4.20/modules.usbmap and/or /etc/hotplug/usb.handmap

Using Bluetooth to access Memory Stick

Once you have either configured a USB Bluetooth dongle or a Bluetooth CF card, you can transfer files between the mobile and the Zaurus wirelessly using the OBEX via bluetooth mechanism. See the bluetooth section for more details.

Controlling the phone with the Zaurus

The K750i also offers a virtual serial interface that can be used to issue AT commands or trigger other phone functions. The serial line is accessible via the USB cable, as well as via Bluetooth. Once the link is established, you can access the data of the phone, interact with the user interface and manipulate the telephony functions.

Serial interface via USB

Once you plug in the phone, the Zaurus will not only recognize the USB storage device, but also detects the built-in modem. It then should load the appropiate cdc_acm module which brings support for USB modems and ISDN adaptors. The new device will be placed under /dev/ttyACM0 and /dev/ttyACM1. The devices can be opened with minicom or any other terminal program to gain access to the phone.

Serial interface via Bluetooth

Instead of relying on a wired connection, you can also use the Bluetooth interface to access this serial line.

Include the following section in the file /etc/bluetooth/rfcomm.conf:

```
rfcomm0 {
bind yes;
# MAC address of your phone:
device 00:00:00:00:00:00;
}
```

Now reload the Bluetooth subsystem with /etc/init.d/bluetooth restart. If you already paired your computer with your phone, accessing the device /dev/rfcomm0 will instruct RFComm to connect to the phone without any user action needed. You can monitor the process with the rfcomm utility: rfcomm

Once a process opens the device file /dev/rfcomm0, the RFComm daemon contacts the phone. See the bluetooth section for more details.

Security

Hardening

- assign passwords to root and zaurus

```
# su
# passwd
# passwd zaurus
```

- tighten down the number of terminals in /etc/securetty

```
console
tty0
tty1
ttyUSB0
```

- disable all non-essential listening ports
  - disable telnet (port 23) if running
  - shutdown ftp server if not used (port 21)
  - shutdown Samba server and portmap (port 111) when not in use
  - shutdown web server when not in use
  - tighten down access for sshd (port 22)
  - disable Qtopia sync with opie-security package (port 4242)
  - disable port 4992 and 4244 with inetd.conf

Firewall

Install iptables modules and configure them as a packet filtering firewall. You can also install shorewall which is a framework that simplifies the management of iptable rules and configuration. To enable IP filtering firewall, install the following:

- iproute - [iproute_z2.2.4-now-ss991023-1_arm.ipk] or [iproute_20010824-1_arm.ipk]
- iptables-base - [iptables-base_2.4.20_arm.ipk]
- iptables-additional - [iptables-additional_2.4.20_arm.ipk] (optional)
- shorewall - [shorewall-c3000_1.4.5-1_arm.ipk]

This version has been customised specifically for the C3000 and C3100, primarily as a firewall while connected to a wireless network. Once installed, you can specify your network interface to be firewalled by modifying */etc/shorewall/interfaces*. The default is to protect the wireless cf (wlan0) network using dhcp.

Once you have established a network connection, you can enable the firewall by issuing the following command:

```
# su
# shorewall start
```

Once you disconnect, you can stop the firewall by issuing the following command:

```
# su
# shorewall stop
```

To check the status of the firewall issue the following command:

```
# shorewall status
```

The iptables-additional package contains extra libraries that will allow you to use all of shorewall's features.

VPN

Setting up a VPN connection with the Z is possible. You will need to install the following packages:

- iproute - [iproute_z2.2.4-now-ss991023-1_arm.ipk] or [iproute_20010824-1_arm.ipk]
- ipsec - [ipsec-module_2.4.20-1_arm.ipk]
- tun - [tun-module_2.4.20-1_arm.ipk]
- vpnc - [vpnc_0.3.2-1_arm.ipk]

Once installed, you can establish a VPN connection by issuing the following command from a console and providing the required settings:

```
# su
# vpnc-connect
```

To terminate the VPN session issue the following command:

```
# su
# vpnc-disconnect
```

sudo

Don't do this on Cacko. This is for SHARP ROM only.

install sudo [sudo_1.6.3p7-2_arm.ipk]

Once sudo is installed, you will need to configure it by using the *visudo* command. It will allow you to update /etc/sudoers

For example:

```
# visudo

    root ALL=(ALL) ALL
    zaurus ALL=(ALL) NOPASSWD: /sbin/ifconfig, /sbin/dhcpcd, /usr/bin/apache,
    /sbin/chroot, /bin/mount, /bin/umount, /sbin/swapon, /sbin/swapoff
```

This uses the security conscious approach of only allowing sudo for the commands you want. If security is not such a concern for you and you just want to be able to sudo any commands as zaurus user without

typing a password, then make it look like this instead:

```
root ALL=(ALL) ALL
zaurus ALL=(ALL) NOPASSWD: ALL
```

Emulators

Game Console Emulators

There are a few emulators for the popular game consoles for the Z. In order to use them, you will need to install SDL libraries [libsdl_1.2.5-slzaurus20050731_arm.ipk] if you have not installed them yet. There is also a nice front end to select and launch the game ROMs. Install the common emulator frontend [zemufe_0.1.1-3ex_arm.ipk]. It also has several addons which allows it to handle additional ROM types and emulators:

- zemufeex-smc_1.0_arm.ipk
- zemufeex-sms_1.0_arm.ipk
- zemufeex-wsc_1.1_arm.ipk

Then install the emulators:

- GameBoy - [zgnuboy_1.0.3-3_arm.ipk]
- Nintendo (NES) - [znester_7.1-1_arm.ipk]
- SuperNintendo (SNES) - [snes9x_sdl-1_arm.ipk]

The following is the key mapping for the emulator:

Key	Action	Key	Action	Key	Action	Key	Action
Enter	START	a	p1 (L)	z	k1 (R)	y u i \ | /	
Space	SELECT	s	p2 (X)	x	k2 (Y)	g h j - -	
Cancel	QUIT	d	p3 (A)	c	k3 (B)	b n m / | \	

You might also need to disable the key repeater while playing the games to avoid stuttering with the sound effects. If you have keyhelper installed, then you can run the following to temporarily disable the key repeater:

```
# khctl norepeat
```

Once you have finished playing the games, you can re-enable the key repeat by running the following command:

```
# khctl repeat
```

psx

You can even emulate Sony Playstation on the Zaurus with the psx4zaurus emulator. It requires the BVDD enabled SDL library. The bundle contains binaries for pdaXrom and Cacko which means you will either need Cacko or Sharp ROM with Tetsu's special kernel as well as the latest BVDD kernel module [bvdd_0.4.0-1_arm.ipk] and libSDL-bvdd [libsdl_1.2.5-bvdd-07-2_arm.ipk] installed.

Extract the psx4zaurus zip file and copy *scph1001.bin* to the same directory where you extracted the *psx4all_cacko* binary. Then place the ROM files into the *games_psx* directory. Use the up and down arrow keys to move between options and the *x* key to select an option.

zbochs

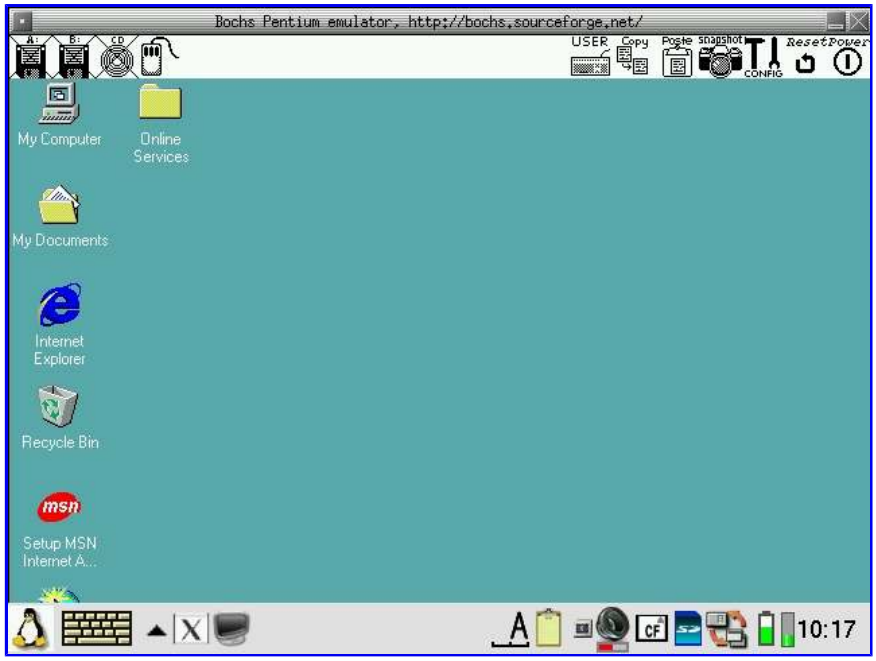
I tried running Win98 on zbochs [zbochs.tgz.gz]. It works but it is extremely slow. Too slow to be much use anyway. However, if you are bend on trying, then get the Linux or Windows version of bochs and use that to create a disk image with at least 250MB (Win98 will fail to install with less than that even though

Microsoft website says it needs less). Then get your Win98 CD-ROM to install Win98 into bochs which will take about 4 hours on a 1 GHz Pentium with 512MB. Once you've done that copy your disk image file, BIOS (BIOS-bochs-latest) and VGA BIOS (VGABIOS-elpin-2.40) to your Zaurus. Also don't forget to copy bochsrc as well and remove the CD-ROM config. Here is a sample bochsrc:

```
romimage: file=BIOS-bochs-latest, address=0xf0000
vgaromimage: VGABIOS-elpin-2.40
megs: 16
ips: 500000
ata0: enabled=1, ioaddr1=0x1f0, ioaddr2=0x3f0, irq=14
ata1: enabled=0, ioaddr1=0x170, ioaddr2=0x370, irq=15
ata2: enabled=0, ioaddr1=0x1e8, ioaddr2=0x3e8, irq=11
ata3: enabled=0, ioaddr1=0x168, ioaddr2=0x368, irq=9
ata0-master: type=disk, path="c.img", cylinders=507, heads=16, spt=63
floppy_bootsig_check: disabled=1
mouse: enabled=1
vga_update_interval: 300000
keyboard_serial_delay: 250
keyboard_paste_delay: 100000
keyboard_mapping: enabled=0, map=
boot: disk
log: bochsout.txt
panic: action=ask
error: action=report
info: action=report
debug: action=ignore
private_colormap: enabled=0
```

I also tried xqt-bochs_2.1.1-1_arm.ipk which appears to be a bit faster than zbochs. Here is Win98 starting up within xqt-bochs:

Eventually, after a long time of waiting, you get this once Win98 has finally loaded:



But really, emulated Win98 runs way too slow, but running DOS inside bochs for playing DOS games is possible.

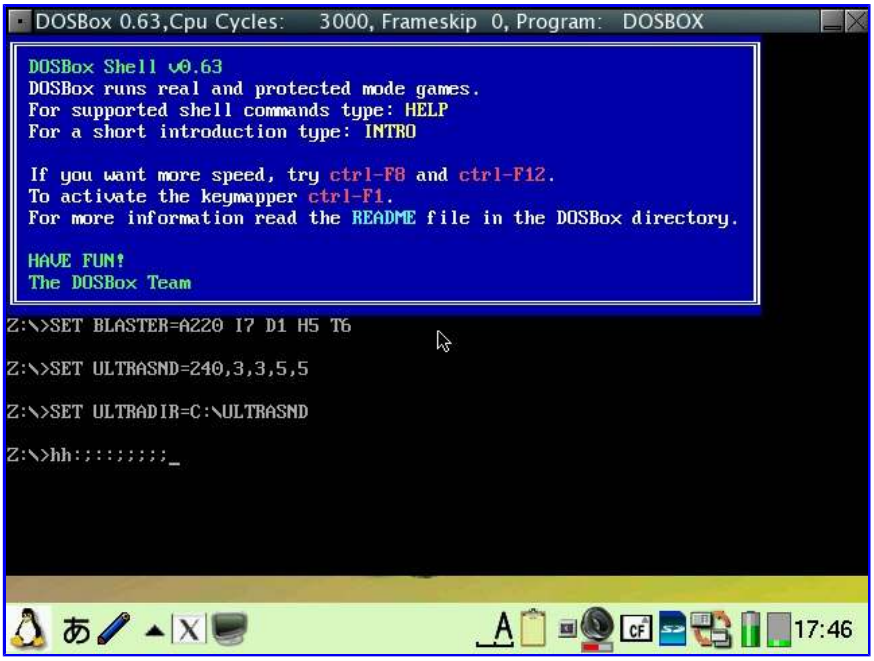
dosbox

dosbox allows you to run DOS applications and games. The advantage over bochs is that it is dedicated and preconfigured to run DOS applications. **chyang** has built a version of dosbox that runs on the C3000 and C3100. I have packaged it as dosbox_0.6.3-3_arm.ipk. Once installed, you can edit /usr/local/dosbox/dosbox.conf and add entries after [autoexec] to automatically launch any DOS application you wish. For example:

```
[autoexec]
mount c /mnt/card/dosfiles
c:
cd tetris
tetris
```

The screen flickers a bit when you launch dosbox. Just press the *Cancel* key a few times. When you have finished using dosbox, press *Shift* + *Ctrl* + *Cancel* to exit.

You might also need to install libstdc6_1.2.2_arm.ipk if you don't already have the standard C libraries installed.


DOSBox 0.63,Cpu Cycles: 3000, Frameskip 0, Program: DOSBOX
DOSBox Shell v0.63
DOSBox runs real and protected mode games.
For supported shell commands type: HELP
For a short introduction type: INTRO
If you want more speed, try ctrl-F8 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>SET ULTRASND=240,3,3,5,5
Z:\>SET ULTRADIR=C:\ULTRASND
Z:\>hh:::::::::_
17:46

Alternatively, you can also install dosbox under X/Qt (or the newer pdaXqtrom package) - [dosbox-x11_0.6.3-3_arm.ipk].

dosbox has a known bug that keys on non-US keyboards are mismatched and some keys are missing, in particular the **:** key. This has been hacked by me for the Zaurus version so that *Fn + ;* is **:**, and *Fn + Ctrl + ;* is **;**

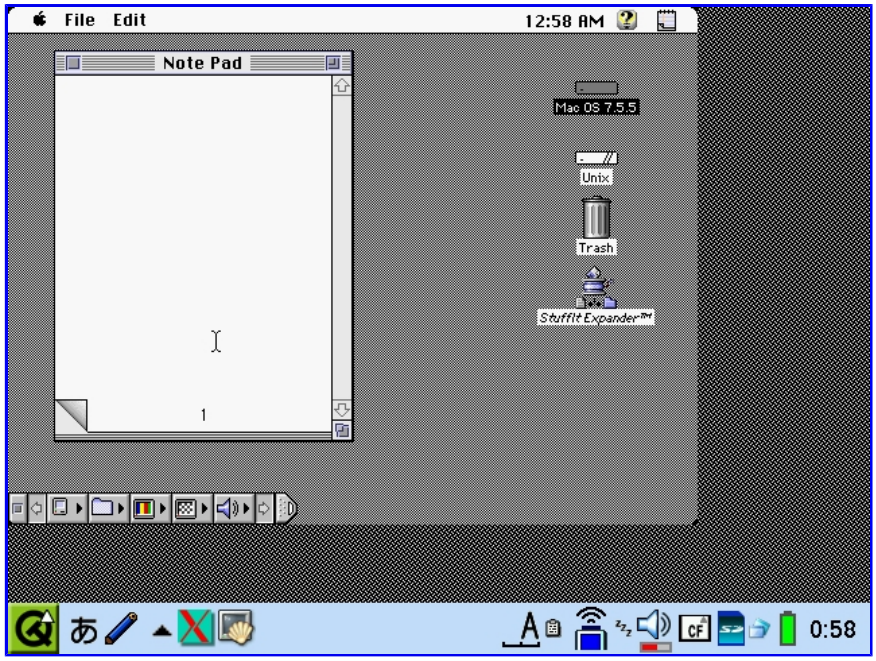
qpose

Emulating the Palm on the Z is possible provided you have a Palm ROM file. Install the following and once done, use the Files tab to locate the ROM file and tap on it. The ROM will then be launched by QPose.

- qpose-data - [qpose-data_3.5-0.2-2_arm.ipk]
- qpose-bin - [qpose-bin_3.5-0.2-1_arm.ipk]

zbasilisk

basilisk [zbasiliskii_0.3_arm.ipk] also works on the Z. Since I already have BasiliskII running on my PC, I simply copied the whole MacOS7.hfv file and the ROM image to the Z and then loaded zbasilisk. It launches X/Qt and runs from within it. zbasilisk emulates the Mac just like basilisk does on the PC, although slower and without sound.

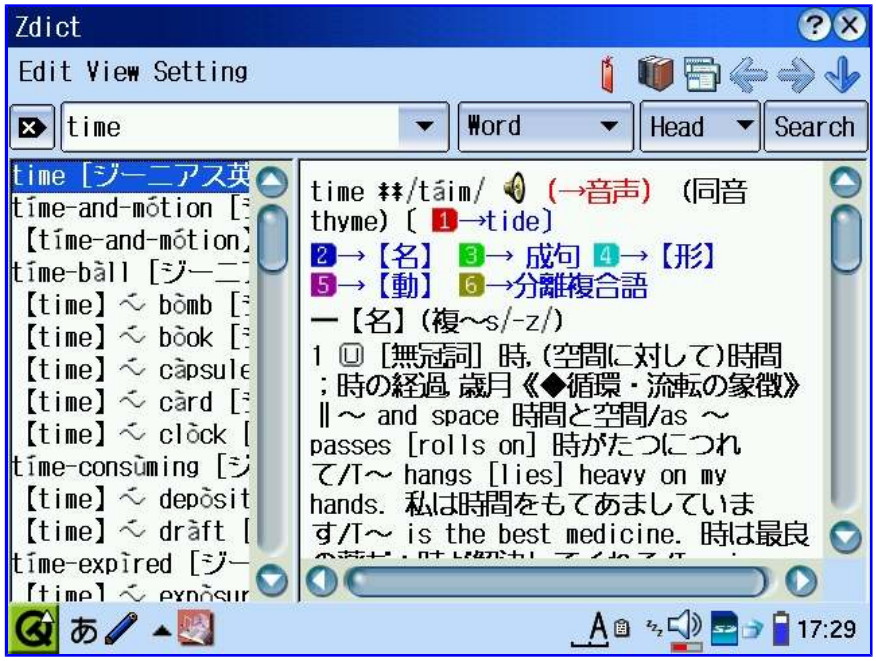


If you want to emulate the Mac on the Zaurus, install BasiliskII on your PC and download the free MacOS7 binaries. Then create a hfv disk image with HFVExplorer (200MB should be enough) and copy the MacOS7.5.3 installer binaries onto the disk image. Then create a MacOS boot disk and start basilisk booting from the floppy. Once booted, install MacOS7.5.3. You might then want to upgrade to MacOS 7.5.5

Dictionaries

The Zaurus already comes with a Japanese-English dictionary which is great, however, it only does Japanese/English and English/Japanese. I need more languages such as German, French and Chinese.

Zdict is the dictionary that comes with the Zaurus. It seems to be almost the same with Zten which is mentioned a lot on the internet. Personally I think Zdict is easier to use than Zten.



Zdict comes with the epwing genius and kojien dictionary packages, but there are some further epwing dictionaries that can be added to either zdict or zten. The following are the ones I have chosen to add:

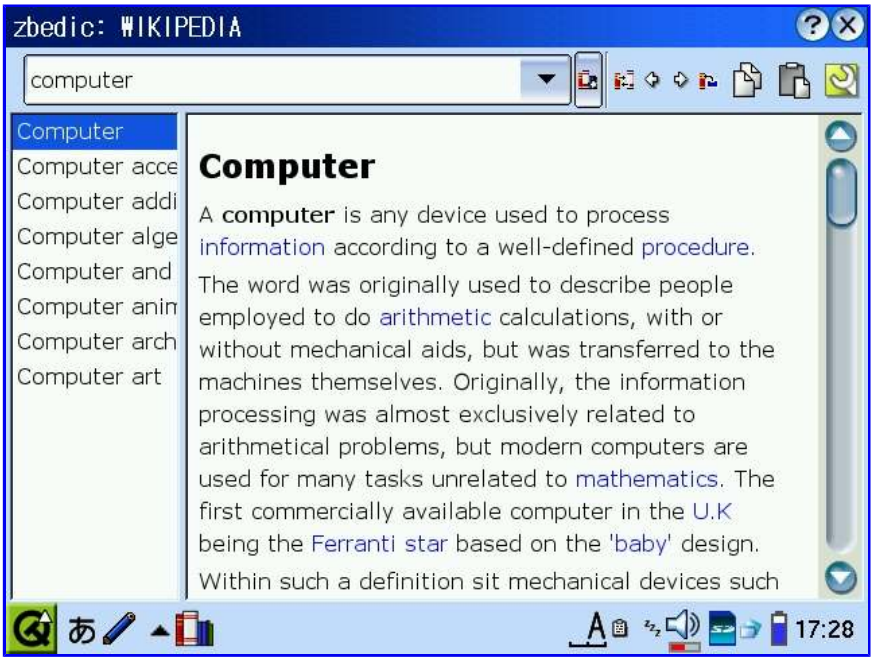
- foldoc [foldoc-fpw1.0.1.zip] - computer terms dictionary
- jarg [jarg-fpw1.2a.zip] - jargon dictionary
- fumeikai [Fumeikai-1.0.zip] - abbreviations dictionary (in English and Japanese)

- wordnet [wordnet-1.6-fpw1.1.3.zip] English dictionary
- kanjidic [kanjidic_en.fpw.tar.gz] kanji dictionary
- wadoku [wadoku-fpw1.1.tar.gz] Japanese-German dictionary
- edict [edict_en.fpw.tar.gz] English-Japanese dictionary

To install these dictionaries, simply extract them to either /hdd3/dict1 or /hdd3/dict2 and select the book from inside the zdict config menu.

ZBedict [zbedic_0.9.4-0_arm.ipk] is another dictionary package that has a large amount of dictionaries in various languages. It also has a huge wikipedia [en-wikipedia.dic.dz]. The following are dictionaries I installed for it:

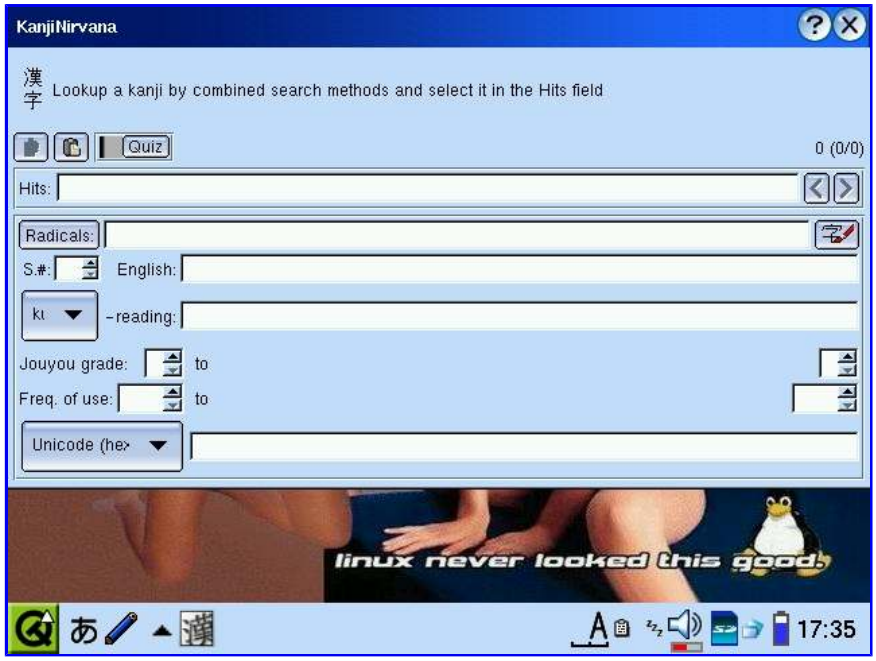
- English-English [en-0.9.0.dic.dz]
- German-English [deen-0.9.0.dic.dz]
- English-German [ende-0.9.0.dic.dz]
- French-English [fren-0.9.0.dic.dz]
- English-French [enfr-0.9.0.dic.dz]
- Spanish-English [esen-0.9.0.dic.dz]
- English-Spanish [enes-0.9.0.dic.dz]
- Italian-English [iten-0.9.0.dic.dz]
- English-Italian [enit-0.9.0.dic.dz]
- Chinese-English [zhen-0.9.0.dic.dz]
- English-Chinese [enzh-0.9.0.dic.dz]
- Japanese-English [jaen-0.9.0.dic.dz]


zbedic: WIKIPEDIA
computer
Computer
Computer acce
Computer addi
Computer alge
Computer and
Computer anin
Computer arch
Computer art
Computer
A computer is any device used to process information according to a well-defined procedure.
The word was originally used to describe people employed to do arithmetic calculations, with or without mechanical aids, but was transferred to the machines themselves. Originally, the information processing was almost exclusively related to arithmetical problems, but modern computers are used for many tasks unrelated to mathematics. The first commercially available computer in the U.K being the Ferranti star based on the 'baby' design.
Within such a definition sit mechanical devices such
17:28

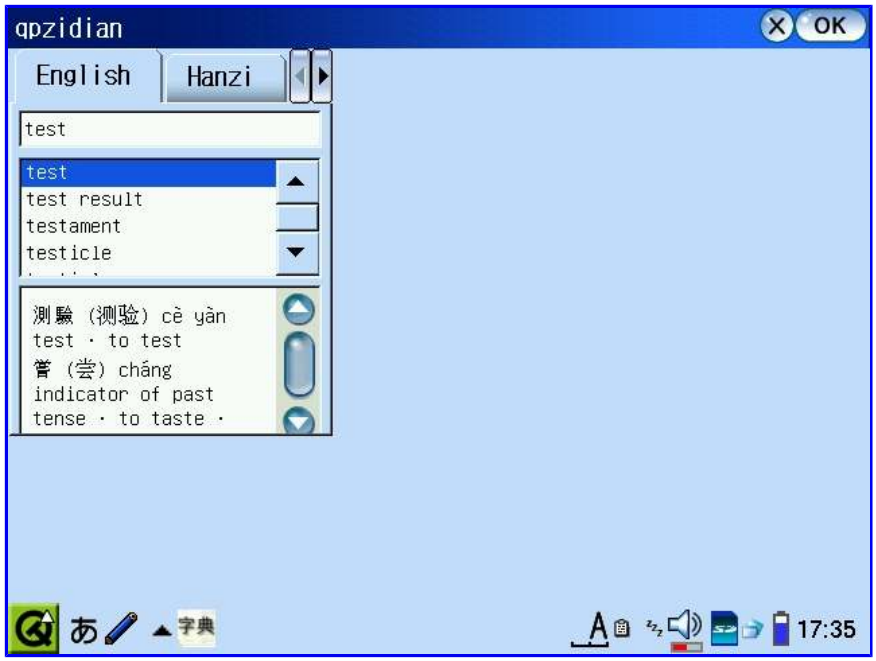
Place the files under a directory such as /hdd3/Documents/dictionaries and do an autodetect in zbedic to locate them.

KanjiNirvana [kani_1.2.0_arm.ipk] is a kanji dictionary and practice tool. The following needs to be done after installation in order to update and add entries:

```
# su
# chown -R zaurus:qpe /home/Qtopia/kani
```



qpzidian [qpzidian_0.1_arm.ipk] is a Chinese/English - English/Chinese dictionary that allows you to look up words via Hanzi and Pinyin.



babbletower [babbletower_0.9.3_arm.ipk] is another dictionary reader written in Java. You need to install one of the J2ME implementations for Zaurus (jeode or personal profile) before you can use babbletower. It is also recommended to install jlauncher [jlauncher_0.1_arm.ipk] after you have installed one of the J2ME implementations so it helps babbletower launch the right J2ME runtime.

Speech Synthesis

flite

flite [flite_arm_bin.tar.gz] will read a text file with a male Scottish voice. To install it, simply gunzip and then untar the files to /usr/local/bin

```
# zcat flite_arm_bin.tar.gz | tar xvf - -C /usr/local/bin
```

I've written a script *saytime* which gets the system time and passes it to flite_time. Also, I am planning to write a GUI interface (using J2ME) for flite [zflite-gui_0.3_arm.ipk] which lets you select a text file and

then it will call flite with the text file as an argument or lets you type in some text and passes it to flite. In the meantime, I have used opie-sh to do something similar [zflite-gui_0.1_arm.ipk] although not as sophisticated and [zflite-gui_0.2_arm.ipk] which uses qshdlg.

Some flite options:

- --sets join_type=simple_join (use simple concatenation of diphones without prosodic modification)
- --setf duration_stretch=1.5 (make it speak slower)
- --setf int_f0_target_mean=145 (make it speak with higher pitch)
- -t "some text"
- -f filename

mbrola

mbrola [mbr301h.zip] is another voice synthesis application. It is an engine that converts diphones (.pho files) to voice (.wav files) and has exchangable libraries for different voices and languages. However, it needs additional software that converts text to .pho files. FreeTTS is such an application, however, it is written for Java 1.4 which does not exist for the Sharp distro (only Java 1.3 is available for Zaurus).

Unfortunately, this means that mbrola is pretty useless until FreeTTS has been backported to Java 1.3 which is not an easy task because FreeTTS uses a lot of the 1.4 features not available in 1.3, or until jamvm which is a jre 1.4 capable java runtime replacement can be made to run on the Sharp distro.

Video Conversion

The Movie Player that comes with the Zaurus is nice for playing MPEG files. I can just copy a .dat file from a VCD and rename it to .mpg and Movie Player can play them in a window and full screen.

However, since space is limited on the Zaurus, (yep, 4GB is nothing if you put a few vids on it), compressing the vids is a good idea. Unfortunately, Movie Player only plays MPG files.

No worries, mplayer will do the job. mplayer can play almost any format and there are also some nice GUI interface for mplayer such as kino2 and zplayer.

Here is what I do using VirtualDub to compress videos (there is a mod of virtualdub that can open mpg files as well - VirtualDub-MPEG2 1.5.10).

From the Virtual Dub Menu:

```
select Video
    Filters...
        add resize
            New width: 320
            New height: 240
            Filter mode: Nearest Neighbour
        Color Depth...
            Decompression format
                16-bit (HiColor/32K)
            Output format to compressor/display
                16-bit (HiColor/32K)
        Compression...
            Microsoft MPEG-4 Video Codec V3
                Configure
                    Control: 30
                    Data Rate: 150
select Audio
    Full processing mode
    Compression...
        MPEG Layer-3
            24 kBits/s,12000Hz,Stereo
```

Then Save as AVI.

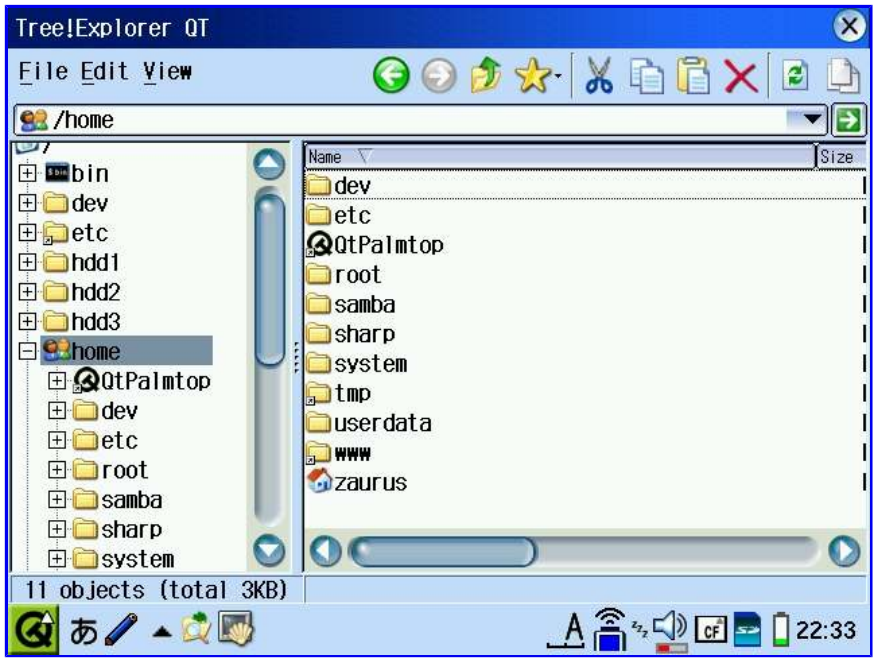
Alternatively, you can also use **PocketDivXEncoder** which is a derivative of mencoder to compress movies for the Zaurus. The default settings with 2-pass encoding does a pretty good job on the compression and quality. You can further tweak the settings for further compression at the loss of some quality.

File Managers

The default File Manager (the Files tab) on the Zaurus is rather limited in functionality. There are, however, some better packages out there.

Tree!Explorer QT

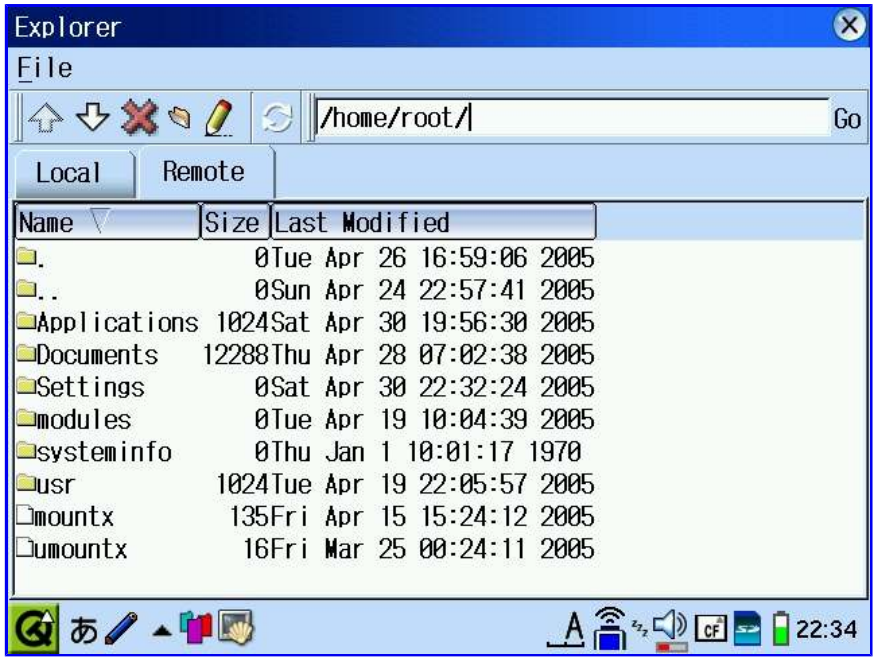
[treeexplorer_1.7.0-2_arm.ipk] and [treeexplorer-p_1.7.0-2_arm.ipk]



This is a nice application similar to Windows Explorer in tree mode with OpenWith and SendTo functionality, however, you only get the full functionality if you pay for the pro edition. The lite version is only good for moving files around but you cannot launch any files. You also get an additional file editor with the pro edition.

Explorer

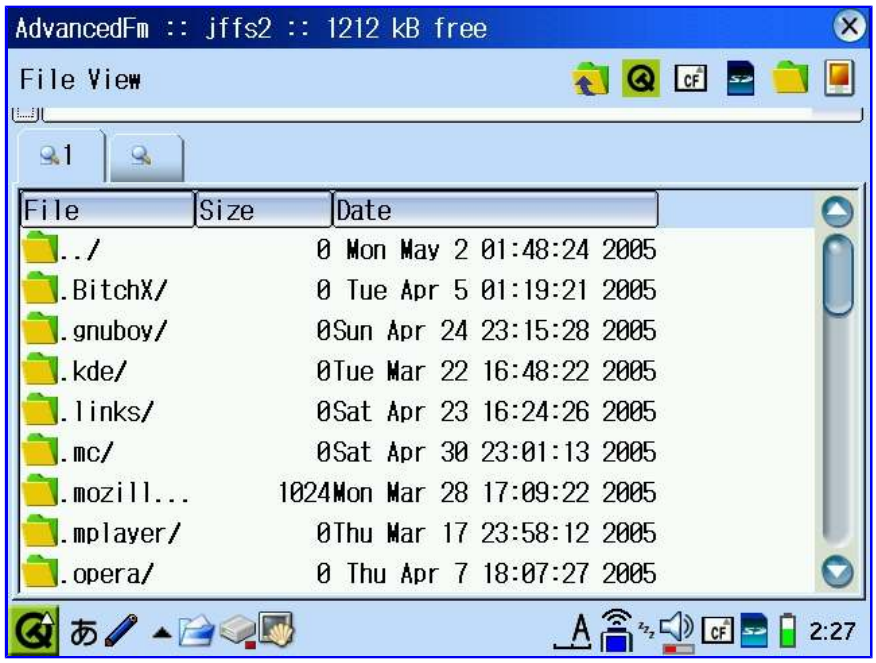
[explorer_1.0_arm.ipk]



This application is for moving and copying files around only. The good thing about it is that it contains two tabs which represent two different directory locations between which files can be easily moved. It also has a FTP client so you can also move files between your local disk and an FTP server. However, it hangs if you copy or move large files.

AdvancedFM

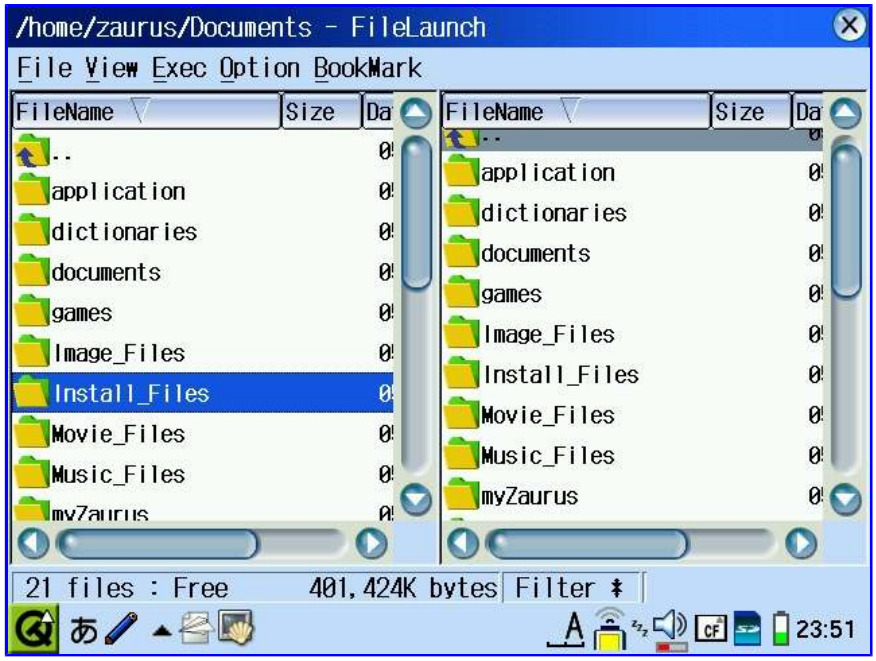
[qtopia-advancedfm_1.0_arm.ipk]


AdvancedFm :: jffs2 :: 1212 kB free
File View
File
Size
Date
../ 0 Mon May 2 01:48:24 2005
.BitchX/ 0 Tue Apr 5 01:19:21 2005
.gnuboy/ 0Sun Apr 24 23:15:28 2005
.kde/ 0Tue Mar 22 16:48:22 2005
.links/ 0Sat Apr 23 16:24:26 2005
.mc/ 0Sat Apr 30 23:01:13 2005
.mozill... 1024Mon Mar 28 17:09:22 2005
.mplayer/ 0Thu Mar 17 23:58:12 2005
.opera/ 0 Thu Apr 7 18:07:27 2005
2:27

This application is almost like the Explorer application mentioned above, but it does not have the FTP functionality. It has, however, a handy bookmark feature for quickly accessing frequently used directories and can open files as text files and run them (if they have appropriate file associations).

FileLaunch

[filelaunch_0.4.5_arm.ipk]



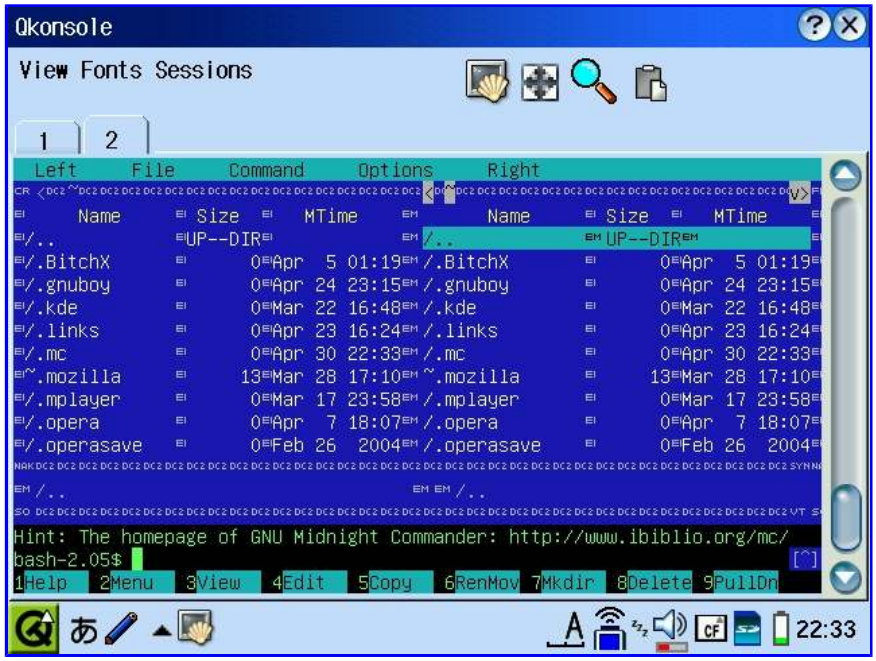
This application is great and heavily customisable. You can have a single explorer panel or split the screen into two panels either horizontally or vertically. It has a full set of file management features and even has options to run each command in sudo mode. There is also a build-in menu system which can be customised to have all your favourite apps in one handy location. It also lets you add your own shell commands in any combinations you want and has already got a set of useful shells for compressing and uncompressing files as well as search and convert. There is also a handy preview feature to view images as well as text and html files.

FileLaunch is originally written in Japanese. I am in the process of translating all the menus and messages to English. The repackaged version [filelaunch-en_0.4.5_arm.ipk] will be available once I finish the translation.

FileLaunch is based on TinyViewer and qshdlg. TinyViewer [tinyviewer_0.3.1_arm.ipk] is a small and simple file browser that allows you to preview files such as images and simple text. tvtools [tvtools_0.0.2_arm.ipk] is an addon to tinyviewer to allow you to view archive files such as zip files.

MidnightCommander

[mc_4.6.0_arm.ipk]


Qkonsole
View Fonts Sessions
1 2
Left File Command Options Right
Name Size MTime Name Size MTime
/.. UP--DIR /.. UP--DIR
/.BitchX 0 Apr 5 01:19 /.BitchX 0 Apr 5 01:19
/.gnuboy 0 Apr 24 23:15 /.gnuboy 0 Apr 24 23:15
/.kde 0 Mar 22 16:48 /.kde 0 Mar 22 16:48
/.links 0 Apr 23 16:24 /.links 0 Apr 23 16:24
/.mc 0 Apr 30 22:33 /.mc 0 Apr 30 22:33
~.mozilla 13 Mar 28 17:10 ~.mozilla 13 Mar 28 17:10
/.mplayer 0 Mar 17 23:58 /.mplayer 0 Mar 17 23:58
/.opera 0 Apr 7 18:07 /.opera 0 Apr 7 18:07
/.operasave 0 Feb 26 2004 /.operasave 0 Feb 26 2004
Hint: The homepage of GNU Midnight Commander: http://www.ibiblio.org/mc/
bash-2.05$
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn
22:33

This is a nice console application like Norton Commander. It has all the file management features that one would need and also works with FTP. However, it uses the extended character set to draw the borders which does not get properly mapped with the Japanese locale unless you use the unismall font which makes the display too tiny.

PIM

The default calendar and addressbook applications aren't that good, so it is better to add something more useful such as kdepim [kdepim_2.1.2_for_SharpRom.ipk.zip] or later, which has a whole suit of PIM applications. PIM software seems to be the most popular and get updated quite frequently, so make sure to get the latest version.

Download and extract kdepim_2.1.2_for_SharpRom.ipk.zip and then install kmicrokdelibs [kmicrokdelibs_2.1.2_arm.ipk] and pimTABicon [pimTABicon_2.1.2_arm.ipk] first. Following this, do the following:

```
# mkdir /hdd3/zaurushome/kdepim
# cd /home/zaurus
# ln -s /hdd3/zaurushome/kdepim kdepim
```

Now install whichever of the following packages you want:

- **OM/Pi email [kopiemail_2.1.2_arm.ipk]**

  You will need to install lib openssl [openssl_0.9.7d_arm.ipk] and [download sr-character-conversion_SharpROM_arm.ipk.zip] for character conversion from the sourceforge project site as well.

- **KO/Pi calendar [korganizer_2.1.2_arm.ipk]**

  You might also want to add [korganizer-alarm_2.1.2_arm.ipk] which provides an alarm applet that will wake the Zaurus from suspend if an event triggers occurs and sounds an alarm.

- **KA/Pi addressbook [kaddressbook_2.1.2_arm.ipk]**

- **PwM/Pi password manager [pwmanager_2.1.2_arm.ipk]**

- **ksharpPIM-DTMaccess [ksharpPIM-DTMaccess_2.1.2_arm.ipk]**

  This is used to sync KA/Pi and KO/Pi with the Sharp PIM applications on the Zaurus, which use the new Sharp DMT Pim format.

- **kmobilephoneaccess [mobilephoneaccess_2.1.2_arm.ipk]**

  Command line tool for accessing mobile phones. It is used from Kx/Pi to sync with / export to mobile phones.

I don't use PIM stuff much and I don't sync either. My Zaurus is my laptop. All my important stuff is on my Z, so I only tried the email, calendar and address book without the rest of the sync stuff. Firefox, Thunderbird and Open Office is what I use mainly.

Java

Java seems to have been discontinued for the Zaurus. The SL-C3000 as well as any later models such as the SL-C3100 do not get shipped with any JVM/JRE (Java Runtime) anymore. Also there has not been any newer versions of the JVM for Zauri since the 1.3.1 release. Java currently is at 1.5.x (April 2005).

There are several Java flavours available for the Zaurus. Most of them are J2ME distributions. J2ME is the micro-edition of Java which was designed for small portable devices.

There is a J2SE package available as well. J2SE is the "normal" Java that runs on most desktop and

laptop computers.

You will need a J2ME flavour of Java to run most of the Java applications available for the other Zauri and PDAs, and a J2SE if you want to try running Java applications that run on PCs.

The following is a small list of available Java runtimes:

- Insignia's Jeode (J2ME that comes with earlier Zauri)
  - on the Zaurus CD - [jeode_1.10.7_arm.ipk]
  - downloadable from Sharp website - [5500v31c.zip]
- Sun's J2ME preview
  - [personal-profile-for-zaurus_arm.ipk]
  - [pp4zaurus-1_0-ea4a-linux-arm-OptimizedJIT_nosym.zip]
  - [java_slc3000_arm.ipk]
- Blackdown Java-Linux (J2SE port for Linux)
  - [java1.3_1.01-oxy2_arm.ipk]
  - [j2re-1.3.1-RC1-linux-arm.tar.bz2]
  - [blackdown-jdk_1.3.1_arm.ipk]

J2ME

Just install one of the J2ME implementation listed above and you should be fine for most Java apps for the Zaurus.

It is technically also possible to run Swing applications with the J2ME distributions using the SwingZ library that you can add to Jeode or the Personal Profile. Simply include the swingz.jar into the classpath. However, this swing library is implementing swing 1.1 only. Most swing applications therefore won't run.

I have also created jlauncher [jlauncher_0.1_arm.ipk] which is a wrapper for evm (jeode) and cvm (personal profile) so that applications specifically packaged for either of those J2ME runtimes can be seemlessly run no matter which of the two J2ME bundles you have installed. For example, a java package bundled for jeode won't run if you only have personal profile installed unless you manually change the script for the application to use personal profile and vice versa. jlauncher will automatically handle it for you.

MIDP

If you also want to run Java applications that run on your mobile phone, ie MIDP flavour of J2ME then you need to install me4se and the midp game and media classes (which you can extract from the midp 2.0 developer kit.

The following are required:

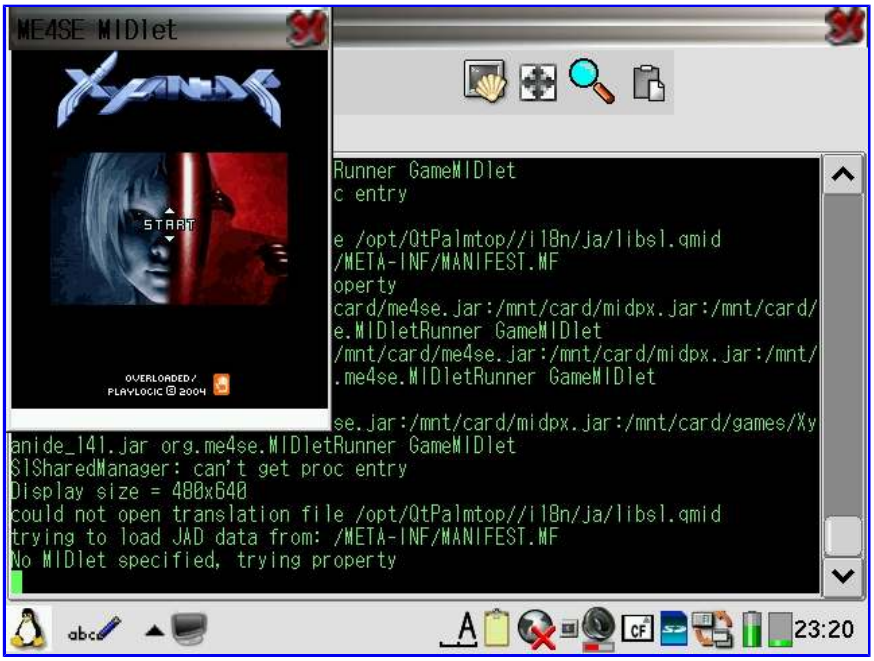
- evm (jeode_1.10.7_arm.ipk)
- me4se.jar
- midp-2_0-src-linux-i686.zip

Extract at a minimum the following files from midp-2_0-src-linux-i686.zip and put them into a zip file retaining the directory structure:

- javax/microedition/lcdui/game/GameCanvas.class
- javax/microedition/lcdui/game/GameDeviceCaps.class
- javax/microedition/lcdui/game/Layer.class
- javax/microedition/lcdui/game/LayerManager.class
- javax/microedition/lcdui/game/Sprite.class
- javax/microedition/lcdui/game/TiledLayer.class
- javax/microedition/media/Control.class
- javax/microedition/media/Controllable.class
- javax/microedition/media/Manager.class
- javax/microedition/media/MediaException.class
- javax/microedition/media/Player.class
- javax/microedition/media/PlayerListener.class
- javax/microedition/media/control/ToneControl.class
- javax/microedition/media/control/VolumeControl.class

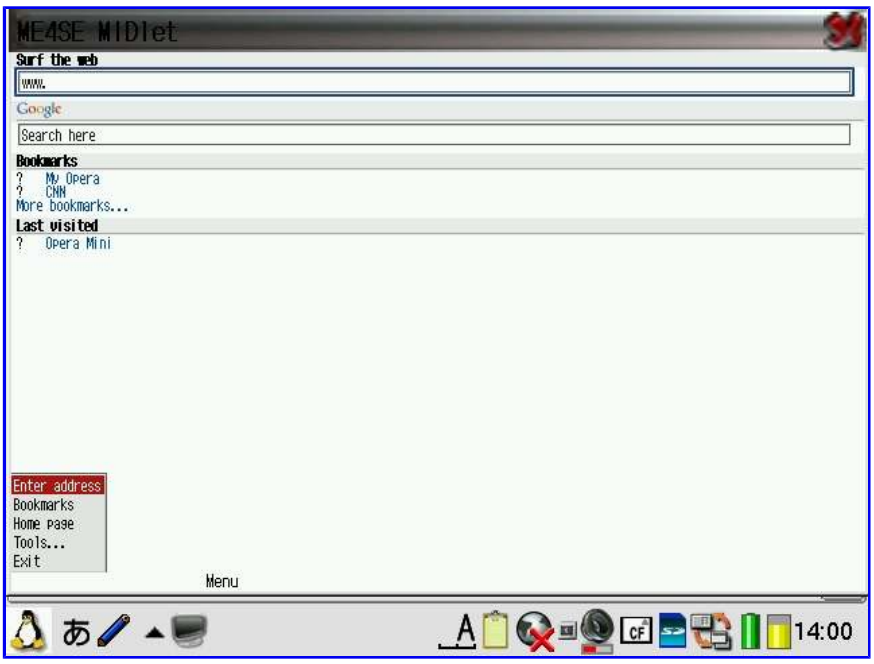
Rename the zip file to something like midpx.jar and then use the following command to launch the mobile phone java game assuming all the jar files are in the current directory and we have a game Xyanide_141.jar:

```
# evm -classpath ./me4se.jar:./midpx.jar:./Xyanide_141.jar org.me4se.MIDletRunner GameMIDlet
```



If you want to play another game then just specify the jar file of that game instead. The last parameter is the classname for the game. This value can be extracted from the jad file or the manifest file inside the jar file and is the last argument of the MIDlet-1 variable after the icon name.

I have also created midp-launcher [midp-launcher_0.2_arm.ipk] which is a qshdlg script providing a GUI game selector to launch game files located under /home/zaurus/Documents/games/j2me. It also includes a modified version of me4se.jar that uses the larger 640x480 screen instead of the default mobile screen size. It also includes other necessary files such as midpx.jar.



midp-launcher also has a command line interface which you can use to launch midp jar files:

```
# midp /mnt/card/opera-mini1.2.2960-basic-us.jar
```

J2SE

Installing the J2SE version from Blackdown is a bit trickier. If you install the ipk version, then you need to do the following hacks to get it working after the install:

```
# su
# ln -s /usr/lib/jdk1.3 /usr/lib/jre
# ln -s /usr/lib/jdk1.3/bin/armv4l /usr/lib/jdk1.3/bin/armv5tel
# ln -s /usr/lib/jdk1.3/lib/armv4l /usr/lib/jdk1.3/lib/armv5tel
# ln -s /usr/lib/libstdc++-3-libc6.1-2-2.10.0.so /usr/lib/libstdc++-libc6.2-2.so.3
# vi /usr/lib/jdk1.3/bin/java
```

replace: APPHOME=`dirname "${0}"`/..
with: APPHOME=`dirname "${0}"`/../lib

replace: prog="${APPHOME}/bin/${proc}/${ttype}/${progname}"
with: prog="${APPHOME}/jre/bin/${proc}/${ttype}/${progname}"

For the bz2 compressed version, you should extract it to /usr/local and then do the following:

```
# ln -s /usr/local/j2re1.3.1/bin/java /usr/bin/java
# ln -s /usr/local/j2re1.3.1/bin/armv4l /usr/local/j2re1.3.1/bin/armv5tel
# ln -s /usr/local/j2re1.3.1/lib/armv4l /usr/local/j2re1.3.1/lib/armv5tel
# ln -s /usr/lib/libstdc++-3-libc6.1-2-2.10.0.so /usr/lib/libstdc++-libc6.2-2.so.3
```

If you don't have the libstdc++ library then install this libstdc6_1.2.2_arm.ipk package.

Remember that if you want to run Java (J2SE) in graphics mode, ie awt and swing, then you need to install X/Qt (see X/Qt section) and you might need additional libraries which may or may not have been bundled with the blackdown distribution you installed. The following is a list of libraries you might need to add (which are contained in additional-ipaq-stuff.tar.gz):

- libBrokenLocale-2.2.2.so linked to libBrokenLocale.so.1
- libXm.so.2.1 linked to libXm.so.2
- libXp.so.6.2 linked to libXp.so.6

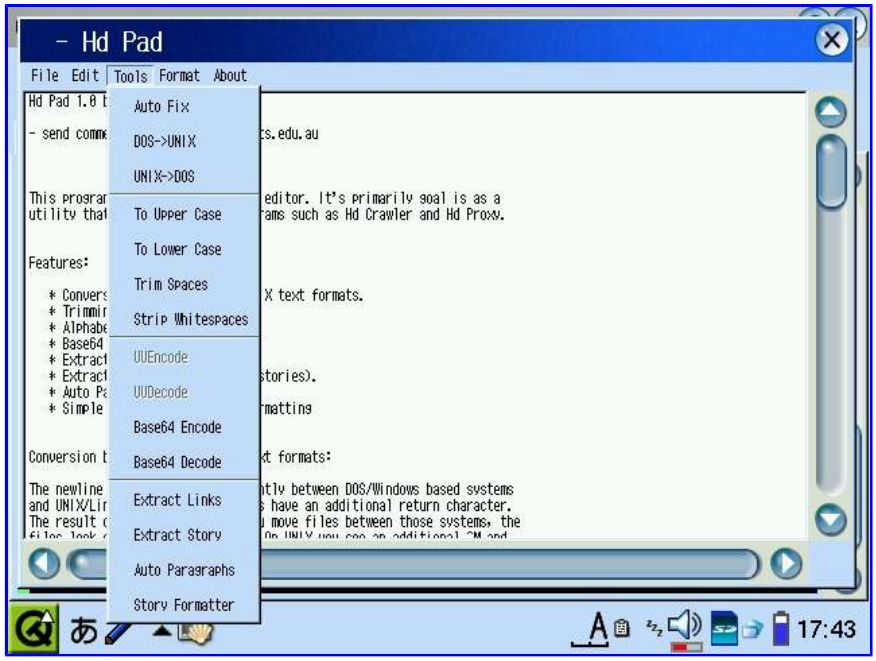
You should also install JSSE which will add SSL support for Java 1.3.x. Download the JSSE zip file (jsse-1_0_3_03-gl.zip) from Sun's website and extract the three .jar files and copy them to the .../jre/lib/ext directory.

If you want to write Java applications on the Zaurus as well in addition to just running them, then you need a Java compiler. The following compilers are available:

- IBM's Jikes - [zaurus_jikes.tar.gz]
- Kopi Compiler - [Kopi.1.5.zip]
- Sun's javac - [tools.jar] from a 1.3.1 JDK

I have created ipk files for the following Java applications that I've written:

- HdPad 1.0 (Java Text Editor) - [hdpad_1.0_arm.ipk] (works with J2ME but needs rt.jar from blackdown; works with J2SE in X/QT)
- HdProxy 1.1 (Java Proxy Server) - [hdproxy_1.1_arm.ipk] (works with J2SE in X/Qt)
- HdCrawler 2.5 (Java Download tool for Yahoo and MSN groups) - [hdcrawler_2.5_arm.ipk] (works with J2SE and JSSE added in X/Qt)



More info about these Java apps can be found at the HdLSoft site.

I have also created a cramfs image with the jre and tools pre-installed and configured. All you need to do to use it is to mount the cramfs image and create some links to the executables by running java-setup.

```
# su
# mkdir -p /mnt/java
# mount -o loop /hdd3/java.cramfs /mnt/java
# echo "/hdd3/java.cramfs /mnt/java cramfs loop 0 0" >> /etc/fstab
# /mnt/java/java-setup
```

See the X/Qt jumbo section for more details. You will need X/Qt if you want to run awt or swing.

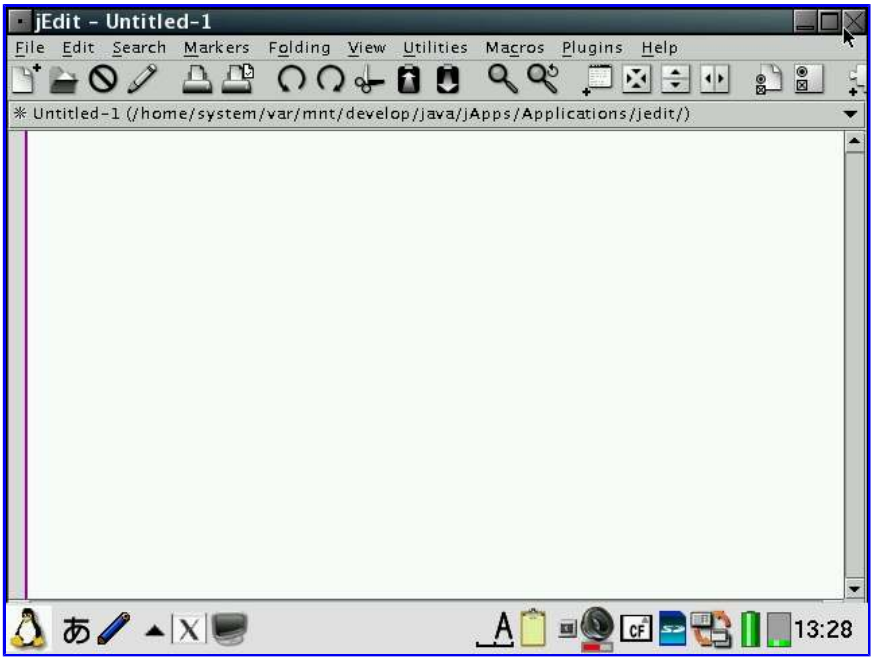
Note: You should install automounter [automounter-c3000_0.5.0_arm.ipk] which automates the creation of additional loop devices and mounting of the cramfs images.

The java cramfs also has HdPad and HdCrawler pre-installed as well as kopi and jikes compiler, and tools from the GNU classpath project.

Java Alternatives

There is a way to run Java 1.4.x applications on the Zaurus through a Java compatible runtime such as JamVM. JamVM is not officially endorsed nor certified to be Java compliant, but it is able to run most Java 1.4.x and even 1.5 applications. JamVM uses libraries from the GNU classpath project to run Java applications.

jEdit can be run using JamVM:


jEdit - Untitled-1
File Edit Search Markers Folding View Utilities Macros Plugins Help
* Untitled-1 (/home/system/var/mnt/develop/java/jApps/Applications/jedit/)
13:28

JamVM GUI applications (AWT and Swing) require a X/Qt environment for display.

gcc

If you want to develop and compile your own C/C++ applications then you will need gcc. There are basically two flavours of development for the Zaurus. You can develop on the Zaurus itself using the on-board or native gcc compiler to build your application binaries or use a cross compiler to build your applications from a PC. The on-board development is particular useful for small applications. However, for larger applications, the Zaurus might not be powerful enough unless you want it sitting on a desk compiling your applications continuously for days. In order to use a cross-compiler, you would need a x86 based PC or something that is powerful enough to emulate it.

There are several cross compile toolchains which allow you to compile applications for the Zaurus. One of the easiest to setup and getting started with is the kopsis toolchain (http://kopsisengineering.com/kopsis/SharpZaurusSdkDsl) which uses the DSL (Damn Small Linux) Live CD technology allowing you to boot the CD and have a ready development environment or use the CD image from a x86 emulator such as qemu, bochs or vmware. There is a great vmware image of DSL made by speculatrix. You can download is from http://www.zaurus.org.uk/downloads.html

There are several on-board development images as well, but none had everything I needed, so I built my own on-board development environment based on the zgcc 2.95.2 cramfs image (zgcc2Bin.cramfs) which is derived from the Debian arm distribution. My zgcc development image (zgcc2-95-2-lite) comes in a single cramfs image and includes necessary headers and libraries to compile and build console based applications, Qtopia applications (QT/E 1.5) as well as kernel modules for the 2.4.20 kernel. This image is as small as it gets and does not include documentation.

I also made a bigger image (zgcc2-95-2) which is compressed as a squashfs image instead to save space. This larger image includes X11 headers and libraries and supports compiling X/Qt applications, especially pdaXqtrom (in fact pdaXqtrom was build using it). It can build many of the opensource applications, both console and X based ones while also retaining the ability to build Qt/E 1.5 applications.

In addition, I also build a newer and even bigger image based on gcc 2.95.3 and a host of updated tools including binutils 2.16, autoconf 2.59, automake 1.9.2, coreutils 5.0, diffutils 2.8.1, gawk 3.1.5, grep 2.5, sed 4.0.9, tar 1.15, texinfo 4.8. A patched glibc 2.2.2 is also bundled and used to link against so applications requiring fesetenv/fegetenv can be successfully compiled with this image. Additionally, this image also contains additional headers and libraries as well as tools to build QT 3.3.6 applications under X11.

The zgcc image that I build is very simple to setup. All you need to do is mount the cramfs or squashfs image and run zgcc-config. Here is an example for a squashfs image:

```
# su
# mkdir -p /mnt/zgcc
# mount -o loop /hdd3/zgcc2-95-2.squashfs /mnt/zgcc
# echo "/hdd3/zgcc2-95-2.squashfs /mnt/zgcc squashfs loop 0 0" >> /etc/fstab
# /mnt/zgcc/zgcc-config
# source /mnt/zgcc/zgcc-env
```

Note: You should install automounter [automounter-c3000_0.5.0_arm.ipk] which automates the creation of additional loop devices and mounting of the cramfs/squashfs images.

Your zgcc is now ready and should get automatically mounted after each reboot (provided you installed automounter, otherwise you will need to mount it manually) and the environment should also be set for you automatically each time you start a new terminal session as the zaurus user.

Temporary files during compilation go to /tmp by default, which will give you problems with larger compiles since the Sharp distro has a size of 1MB for /tmp. The easiest way to fix this is to set the TMPDIR variable and point it to somewhere with more space such as /hdd2/tmp.

```
# mkdir -p /hdd2/tmp
# export TMPDIR=/hdd2/tmp
```

The zgcc development image also comes with some simple samples. There is a console helloworld application and a sample Makefile to compile it. I also included a hello-qt sample which demonstrates a simple Qtopia version of helloworld. And last but not least, I also included a sample driver module and Makefile to test building kernel modules.

tmake also works for generating Makefile for Qtopia applications. You can even use **configure** to generate the *Makefile* for compiling source packages for various Linux ports and projects, in particular X/Qt and pdaXqtrom sources. Perl and xml-parser is also bundled with the larger images.

If you su to root user, then the environment variable for gcc are not set automatically. This was done on purpose. If you want to enable gcc for the root user temporarily, then do the following:

```
# source /mnt/zgcc/zgcc-env
```

If you have the 2.95.3 image, then you can also compile Qt 3.3.6 applications. However, the default environment is configured for Qt/E development. To switch to QT 3.3.6, do the following:

```
# source /mnt/zgcc/qt3/qt3-env
```

Note: QT 3.3.6 support is experimental. Not everything might work or compile.

Perl

You do not need to install Perl separately if you have installed the zgcc image, but if only want Perl without the gcc compiler, then do the following:

```
# su
# ln -s /usr/local /home/root/usr
```

Then install the following packages:

- libperl - [libperl_5.6.1_arm.ipk]
- perl-base - [perl-base_5.6.1_arm.ipk]
- perl5 - [perl_5.6.1_arm.ipk]

Once the packages have been installed add the following to the .profile file:

```
export LANGUAGE=C
export LC_ALL=C
```

You might also want to do the following:

```
# su
# ln -s /usr/local/bin/perl /usr/bin/perl
```

Optionally, you might also install a XML parser [xml-parser_2.31-1_arm.ipk] module for perl.

Installing X/Qt

This can be a quite simple and straightforward task or a messy and frustrating experience. If you are lucky, everything just installs and you got X/Qt up and running in no time at all. However, if you are unlucky, then troubleshooting a broken installation can be a challenging and frustrating process. I've sucessfully installed the latest version of X/QT, but I have noticed that some X/Qt packages do not uninstall cleanly and will confuse reinstalls or other packages that have dependancies on them. Not all the packages that I installed are required for basic X to work under Qtopia, however, with this set of libraries, you can easily run Firefox and Debian PocketWorkstation.

You can either follow the instructions below or use the jumbo package instead. The jumbo package section is more up to date.

Install the following packages in the the given order:

- xqt-fonts-misc [xqt-fonts-misc_4.3.0-3_all.ipk]
- xqt-fonts-100dpi [xqt-fonts-100dpi-iso8859-1_4.3.0-3_all.ipk]
- xqt-fonts-75dpi [xqt-fonts-75dpi-iso8859-1_4.3.0-3_all.ipk]
- xqt-server [xqt-server_1.9.0_arm.ipk]
- xbase-etc [xbase-etc_4.3.0-3_all.ipk]
- zlib - [zlib_1.2.2-1_arm.ipk]
- xlibs [xlibs_4.3.0-3_arm.ipk]
- xbase-client [xbase-clients_4.3.0-3_arm.ipk]
- gdk-pixbuf [gdk-pixbuf_0.22.0-2_arm.ipk]
- glib [glib_1.2.10-0vl3_arm.ipk]
- glib-additional [glib-additional_1.2.10-2_arm.ipk]
- glibc-locale-ja-eucjp [glibc-locale-ja-eucjp_2.2.2-1_arm.ipk]
- glibc-locale-ja-utf8 [glibc-locale-ja-utf8_2.2.2-1_arm.ipk]
- glibc-gconv-ja [glibc-gconv-ja_2.2.2-1_arm.ipk]
- glib2 [glib2_2.4.7_arm.ipk]
- libgcc1 [libgcc1-zaurus_3.2.2-0_arm.ipk]
- freetype [freetype_2.1.5-1_arm.ipk]
- fontconfig [fontconfig_2.2.1-1_arm.ipk]
- fontconfig-etc [fontconfig-etc_2.2.1-1_all.ipk]
- xqt-fonts-encodings [xqt-fonts-encodings_4.3.0-3_all.ipk]
- atk [atk_1.6.1_arm.ipk]
- gtk [gtk_1.2.10.1_arm.ipk]
- gtk2 [gtk2_2.4.13_arm.ipk]
- libpng3 [libpng3_1.2.4-1_arm.ipk]
- libtiff [libtiff3.6.1-1_arm.ipk]
- pango [pango_1.4.1_arm.ipk]
- blackbox [blackbox_0.65.0-2_arm.ipk]
- rxvt [rxvt_2.6.4-1_arm.ipk]
- xqtclip [xqtclip_0.0.2_arm.ipk]
- xqt-startup-scripts [xqt-startup-scripts_0.0.3_all.ipk]

Once the packages are installed, do the following:

```
# cd /opt/QtPalmtop/bin
# su
```

```
# ln -s Xqt X
# ln -s rxvt xterm
```

add the following to /home/zaurus/.xinitrc

```
xmodmap -e "keycode 69 = slash comma"
xmodmap -e "keycode 70 = period question"
xmodmap -e "keycode 22 = minus grave"
xmodmap -e "keycode 60 = grave"
```

Hint: you can use the command line program *xev* from within X to determine the keycode of different keys on the SL-C3000 and SL-C3100 keyboard and add the mapping with xmodmap like I did, or alternatively, you can create a .xmodmaprc file to customize your keymaps instead:

```
# xmodmap -pke > /home/zaurus/.xmodmaprc
```

In addition, I modified .xinitrc and replaced the line:

```
rxvt &
```

with the following:

```
if [ -f $HOME/.xstart ]; then

	XAPP=`cat $HOME/.xstart`
	$XAPP &
	rm $HOME/.xstart

else

	rxvt &

fi
```

Then I replaced the content of /home/QtPalmtop/bin/startx-wrapper with the following:

```
#!/bin/sh
X=`ps -ef|grep X|grep qt`
if [ "$X" = "" ]; then

	if [ "$1" != "-qcop" ]; then
		echo $1 > /$HOME/.xstart
	fi
	startx

else

	export DISPLAY=:0.0
	$1 &

fi
```

Create a link:

```
# ln -s /home/QtPalmtop/bin/startx-wrapper /home/QtPalmtop/bin/xlauncher
```

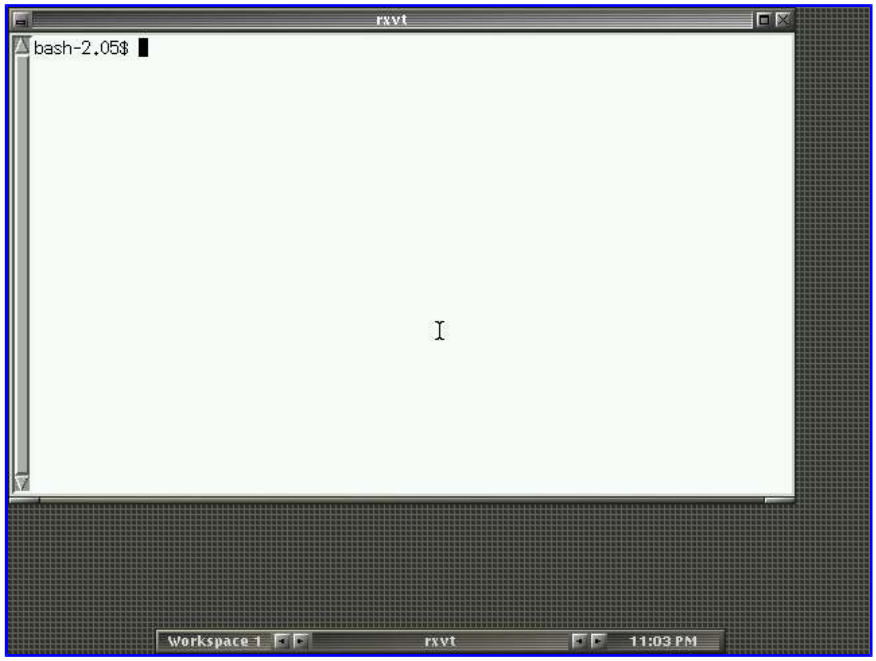
With this modification, I can run X application such as firefox by issuing the command:

```
# xlauncher firefox
```

This causes firefox to be run, and if X is already running it will just appear inside X, but if X has not been started yet, it will start X as well. This is nice for adding X application icons on the Qt desktop.

X control can be activated by pressing the **menu** button which allows you to switch to fullscreen and shutdown X.

Note: You can also install [xqt-debian-scripts_0.5_arm.ipk] instead of xqt-startup-scripts [xqt-startup-scripts_0.0.3_all.ipk] which will give you an even more enhanced .xinitrc and startx-wrapper and you don't need to do the above modifications manually. (It will also give you an addtional X/Qt tab and some pretty icons for X applications such as firefox, the gimp, abiword and pocketworkstation which you can delete if you don't like them)



If you want to try installing Thunderbird or Minimo, then do the following as well:

```
# su
# pango-querymodules > /etc/pango/pango.modules
# mkdir -p /etc/gtk-2.0
# gdk-pixbuf-query-loaders > /etc/gtk-2.0/gdk-pixbuf.loaders
```

You also need to install the following patch:

- xqt-libXrender [xqt-libXrender_1.2.2_arm.ipk]

This will fix the fonts problem with Thunderbird. Without this updated library the fonts in Thunderbird are not being displayed at all.

Debian/PocketWorkstation

You can run Debian in chroot mode so it can coexist with the existing system (Sharp ROM with Qtopia). You will need to install X/Qt first (see above) or run it via VNC. You could also install Debian using my pre-build debian image (see the X/Qt jumbo and PocketWorkstation section), otherwise follow the instructions below to install Debian manually.

Debian needs to be installed on an ext2 filesystem with at least 195MB of free space. Unfortunately, /hdd3 is vfat. You could reformat /hdd3 as an ext2 filesystem (but that is troublesome), or install Debian to an ext2 formatted CF or SD. Alternatively, you can create a loopback filesystem on /hdd3 and format it as ext2.

Since there is space on /hdd3, creating a loopback filesystem for Debian would be the best unless you have already filled up /hdd3 in which case you can install Debian to a SD or CF card.

You can either follow the instructions below or use the jumbo package instead.

If not using the jumbo package, here is how you create the loopback filesystem with 256MB on /hdd3 (if you want to install OpenOffice, make it at least 512MB):

```
# su
# cd /hdd3
# mkdir debroot
# mknod /dev/loop2 b 7 2
# dd if=/dev/zero of=/hdd3/pocketworkstation bs=1M count=256
# echo y | /sbin/mke2fs pocketworkstation
# mount -o loop -t ext2 /hdd3/pocketworkstation /hdd3/debroot
```

Here is how you format your SD card to be ext2 (insert an empty SD card, otherwise you will lose everything that was on it):

```
# su
# umount /mnt/card
# mkfs.ext2 /dev/mmcda1
# mount -t ext2 /dev/mmcda1 /mnt/card
# mkdir /mnt/card/debroot
```

You can also leave your SD as FAT and not format it as ext2. You will need to create a loopback filesystem on your SD similar to the above sample for creating a loopback filesystem on /hdd3.

Also create a 128MB swapfile for Debian if you haven't got one yet.

```
# su
# dd if=/dev/zero of=/hdd3/swapfile bs=1048576 count=128
# mkswap /hdd3/swapfile
```

Now we are ready to install Debian [zaurus-debian-big-v0.17.tgz]. We will now assume installation into /hdd3/debroot. Replace with /mnt/card/debroot as appropriate. Change to the directory where zaurus-debian-big-v0.17.tgz is located and then do the following:

```
# zcat zaurus-debian-big-v0.17.tgz | tar xvf - -C /hdd3/debroot
# su
# cd /hdd3/debroot
# pwd > /etc/debroot
# chown -R root:root etc
# chown -R root:root var
# chown -R root:root home
# mkdir -p mnt/card
# mkdir -p mnt/cf
# cp /etc/hosts etc
# cp /etc/resolv.conf etc
# vi startd
```

```
#!/bin/sh
### startup commands ###
export DISPLAY=0:0
```

```
/usr/bin/icewm-session
### shutdown commands
umount /mnt/card 2>/dev/null
umount /mnt/cf 2>/dev/null
umount /proc
```

```
# chmod 755 startd
# cd home
# ln -s ../root root
# mkdir zaurus
# cd ..
# cp -R INSTALL.d/debroot/root/.icewm home/zaurus
# cp INSTALL.d/debroot/usr/local/bin/* usr/local/bin
# cp INSTALL.d/native/bin/* /usr/local/bin
# cp INSTALL.d/native/debroot.conf /etc
# rm -r INSTALL.d
```

You will also need sudo (see above in sudo section). Once sudo is installed, add the following to the zaurus user's NOPASSWD list using visudo:

- /bin/mount
- /sbin/chroot
- /sbin/swapon

Install [xqt-debian-scripts_0.5_arm.ipk] (unless you installed the jumbo package) and then click on the Debian icon on Qtdesktop. Wait a bit for X and PocketDesktop to load. You won't see the taskbar at first because it is hidden beneath your Qtopia taskbar. Press your menu key to change to fullscreen mode.

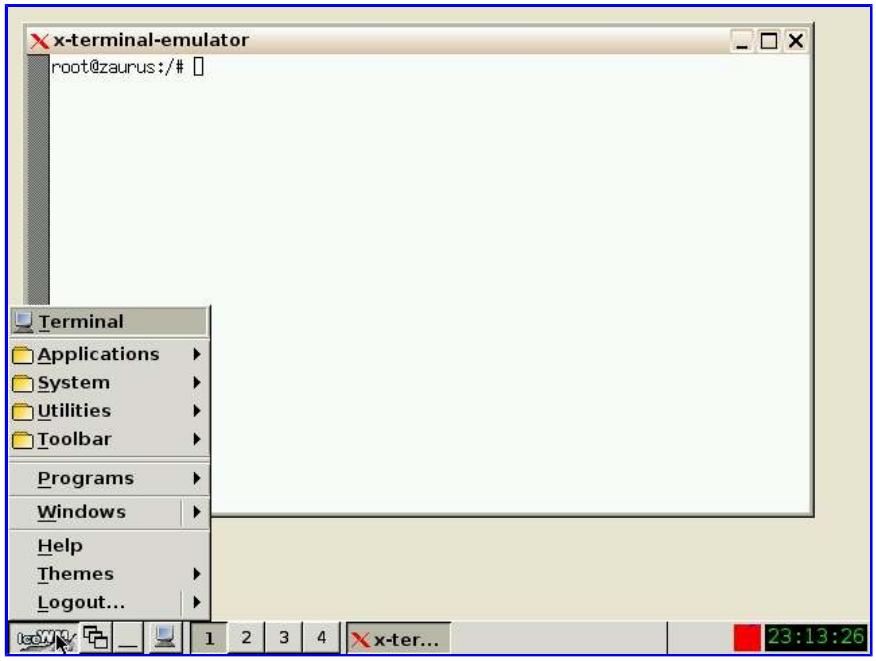
If you want the icewm task bar to appear at the top instead of the bottom of the screen, then edit /home/zaurus/.icewm/settings and change the *TaskBarAtTop* entry.

```
TaskBarAtTop=1 # 0/1
```

If you don't want to use sudo, then you can start PocketWorkstation as root. You need to copy .xinitrc from /home/zaurus to /home/root and then make sure you use **su -** before you run **xlauncher debian** from a console/terminal window. You will also need to copy the .icevm directory. However, the icons on the Qt Desktop cannot be launched as root. You will have to configure sudo if you want to be able to launch using those icons.

```
# su
# cp /home/zaurus/.xinitrc /home/root
# chown root:root /home/root/.xinitrc
# cd /hdd3/debroot/home
# cp -R zaurus/.icewm root
# cd /hdd3/debroot
# su -
# xlauncher debian
```

The xqt-debian-scripts package is an enhanced replacement for xqt-startup-scripts. Your X will work just as before, so you will now have a X server icon and a Debian icon. Your existing X apps will launch into normal X if there is no other X server running, but if Debian is running, they will just launch into the Debian X session and assume the look and feel of icewm. They will look the same as your Debian applications, however, since the applications were launched outside the chroot environment, they retain their access to the normal environment but just appear inside the same X window session. The xqt jumbo package includes the xqt-debian-scripts package, so if have installed the xqt jumbo package, then don't install the xqt-debian package since it is already included.



Use the **xlauncher** command to start X applications from the command line or use it as the launcher for a Qt desktop icon (see Firefox section for an example).

The mouse function is emulated as follows:

- Tapping on screen = Left Click
- Fn + Shift + Tapping = Center Click
- Fn + Tapping = Right Click

The **xlauncher** also automatically mounts and binds your SD and CF cards if they are inserted (before Debian is started) so that they will be acessible from within Debian.

You can also manually mount and unmount them. Here is an example on how to mount your SD card so it can also be used from the chrooted Debian:

```
# su
# mount -o bind /mnt/card /hdd3/debroot/mnt/card
```

The following unmounts the SD card so it is unmounted from the chrooted Debian but still mounted for Qtopia/uLinux environment:

```
# su
# umount /hdd3/debroot/mnt/card
```

Note: There are only 2 loop devices by default. You might need to create more. See filesystem section for more details.

Since there are regular new builds of Debian available, this Debian image will probably be slightly out of date. You can update your Debian installation by running the following commands (assuming you are connected to the net). This step is not really necessary unless you want the latest and greatest (and have sufficient space for whatever gets thrown at you):

```
# source /root/.profile
# apt-get update
# apt-get upgrade
# apt-get clean
```

You can now use apt-get to install new applications. apt-get by default connects to the internet to get the required packages for you. If you already have the required files downloaded or don't want to connect to the internet for installations, then you can also make apt-get install your .deb files from a local directory. You would need to install the dpkg-dev package first.

Once that is done, make a directory such as /home/zaurus/Documents/Install_Files/debs and place all your .deb files in there. Then generate a package summary file (Packages.gz).

```
# mkdir -p /home/zaurus/Documents/Install_Files/debs
# cp *.deb /home/zaurus/Documents/Install_Files/debs
# cd /home/zaurus/Documents/Install_Files
# dpkg-scanpackages debs /dev/null | gzip > debs/Packages.gz
```

now add the following as the first entry in /etc/apt/sources.list

```
deb file:/home/zaurus/Documents/Install_Files debs/
```

apt-get should now look for files located under /home/zaurus/Documents/Install_Files/debs when you try to install a package. You can also comment out the other entries with a hash (#) if you don't want apt-get to connect to the internet. Also, whenever you add new files to /home/zaurus/Documents/Install_Files/debs you will need to re-run dpkg-scanpackages

As an alternative to running Debian under X/Qt, you can also use VNC to run Debian instead, or even do a combination of X/Qt and VNC. The VNC and X/Qt sessions will have different settings, but if you want them to at least share some of the icevm settings, do the following:

```
# su
# cd /hdd3/debroot
# ln -s home/zaurus/.icevm root/.icewm
```

Make sure the DEBROOT in /etc/debroot.conf is set to /hdd3/debroot

```
# The chroot directory for the Debian installation
DEBROOT=/hdd3/debroot
```

To start the VNC server, do the following:

```
# su
# Vncserver
```

Now just connect to it via a vnc client such as keypebble (see VNC section).

OpenOffice 1.1.4

Yes, OpenOffice works on the C3000 and C3100! However, you will need to install PocketWorkstation first which also involves installing X/Qt (see other sections). Alternatively, you can also run OpenOffice as a cramfs image with the X/Qt jumbo package without installing PocketWorkstation (see X/Qt jumbo section for further details). The following section describes installing OpenOffice under PocketWorkstation.

You will also need 200MB of free disk space on your PocketWorkstation disk to store the installed OpenOffice files. In addition, you also need 78MB of space for the extracted installation binaries. The compressed OpenOffice installation binary [OOo_SRX645_linuxarm_install.tar.gz] is 71MB in size. If you have sufficient space, you will be able to install OpenOffice.

If you installed Debian on a 512 MB or greater SD or CF, or on a 512 MB loopback filesystem on hdd3, and have more than 200 MB of free space left on those devices, then you are ready to install OpenOffice. However, if you installed Debian on a 256 MB loopback system or don't have enough free disk space left on your loopback filesystem, but still have plenty of disk space left on hdd3, then you can create and mount an additional loopback filesystem. To create an additional loopback filesystem do the following:

```
# su
```

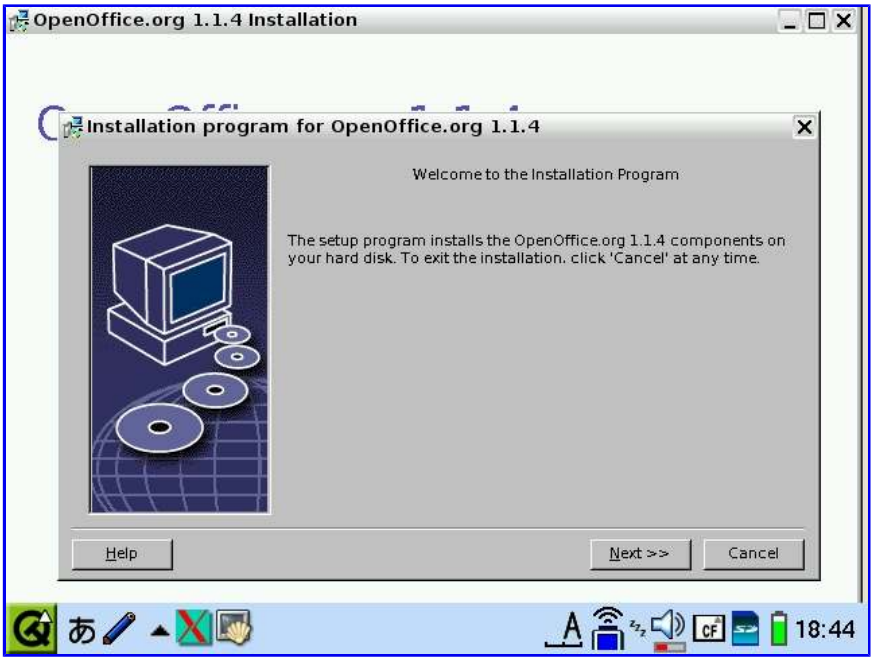
```
# cd /hdd3
# mkdir OpenOffice.org1.1.4
# mknod /dev/loop3 b 7 3
# dd if=/dev/zero of=/hdd3/openoffice bs=1M count=256
# echo y | /sbin/mke2fs openoffice
# mount -o loop -t ext2 /hdd3/openoffice /hdd3/OpenOffice.org1.1.4
# mount -o loop -t ext2 /hdd3/pocketworkstation /hdd3/debroot
# mkdir -p /hdd3/debroot/home/zaurus/OpenOffice.org1.1.4
# mount -o bind /hdd3/OpenOffice.org1.1.4 /hdd3/debroot/home/zaurus/OpenOffice.org1.1.4
```

For preparing for the installation I have extracted the OpenOffice installation binary onto my SD card which can be easily shared between the Qtopia and chrooted Debian system. If you are installing to the SD card, make sure you have sufficient additional space for the extracted files or complement the space with a CF card and vice versa.

```
# zcat OOo_SRX645_linuxarm_install.tar.gz | tar xvf - -C /mnt/card/Documents
```

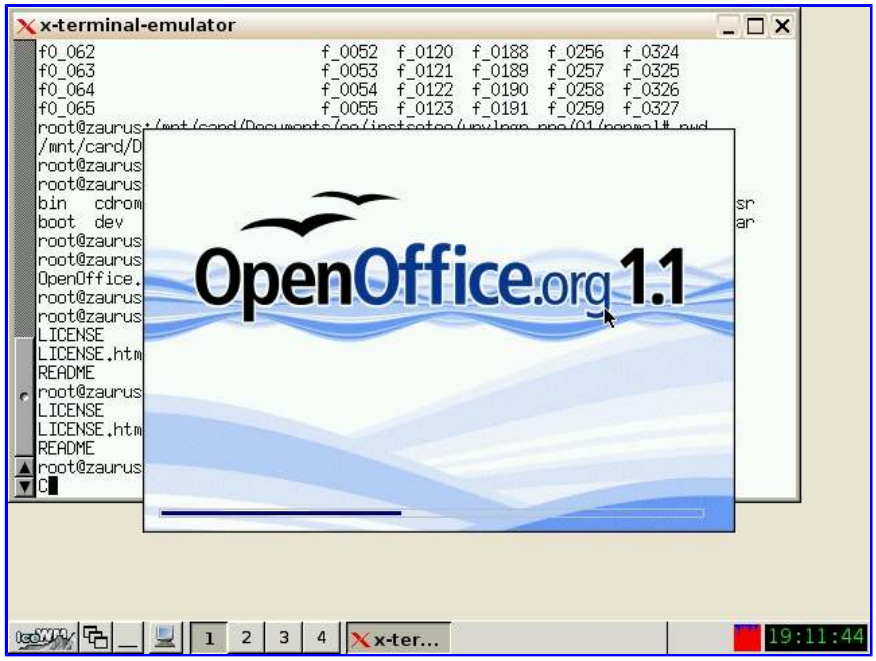
Now start the PocketWorkstation instance and open a terminal window from inside it.

```
# cd /mnt/card/Documents/instsetoo/unxlngr.pro/01/normal/
# ./setup
```


OpenOffice.org 1.1.4 Installation
Installation program for OpenOffice.org 1.1.4
Welcome to the Installation Program
The setup program installs the OpenOffice.org 1.1.4 components on your hard disk. To exit the installation, click 'Cancel' at any time.
Help
Next >>
Cancel
18:44

Wait until the installer launches and install it chosing your desired options. Make sure you install to /home/zaurus/OpenOffice.org1.1.4 which is the default or you would need to change a few things manually to reflect the difference. Once installed you can start OpenOffice as follows:

```
# cd /home/zaurus/OpenOffice1.1.4
# ./soffice &
```



If you see messages complaining about locale, then do the following:

```
# cd /home/root
# echo "export LC_ALL=C" >> .profile
# source .profile
```

You can add OpenOffice to the icevm menu by adding the following into /etc/X11/icewm/programs

```
menu "OpenOffice" folder {
prog "Writer" - sh -c "/home/zaurus/OpenOffice.org1.1.4/program/swriter"
prog "Impress" - sh -c "/home/zaurus/OpenOffice.org1.1.4/program/simpress"
prog "Draw" - sh -c "/home/zaurus/OpenOffice.org1.1.4/program/sdraw"
prog "Math" - sh -c "/home/zaurus/OpenOffice.org1.1.4/program/smath"
prog "Calc" - sh -c "/home/zaurus/OpenOffice.org1.1.4/program/scalc"
}
```

Make sure you are using the latest version of xqt-debian-scripts [xqt-debian-scripts_0.6_arm.ipk] which has additional support for mounting the extra OpenOffice loopback if it exists. If you have installed OpenOffice onto an additional loopback filesystem, you will also need to create /etc/openoffice.conf with the location of OpenOffice stored in it:

```
# echo "/home/zaurus/OpenOffice.org1.1.4" > /etc/openoffice.conf
```

In addition you will need to modify **startd** to look as follows:

```
#!/bin/sh
### startup commands ###
export DISPLAY=0:0
/usr/bin/icewm-session
### shutdown commands
if [ -f /etc/openoffice.conf ]; then

      umount `cat /etc/openoffice.conf` 2>/dev/null

fi
umount /mnt/card 2>/dev/null
umount /mnt/cf 2>/dev/null
umount /proc
```

Installing Mozilla (Firefox and Thunderbird)

Mozilla Firefox and Thunderbird on X/Qt

Make sure you install X/Qt first (either manually, using X/Qt jumbo package or the X/Qt jumbo cramfs image). Once that is done you can just install Firefox and Thunderbird.

To install firefox you need either the original [firefox_0.9gtk_armv5tel.ipk] or the modified [firefox0.9-3_arm.ipk]. If you are using the original package, you need to do the following:

```
# su
# chown -R zaurus:qpe /usr/lib/firefox*
```

If you have a C3000, space on /home is quite scarce so it is better to move the firefox profiles to somewhere with more space.

```
# su
# mkdir -p /hdd3/zaurushome
# mv /home/zaurus/.mozilla /hdd3/zaurushome
# ln -s /hdd3/zaurushome/.mozilla /home/zaurus/.mozilla
```

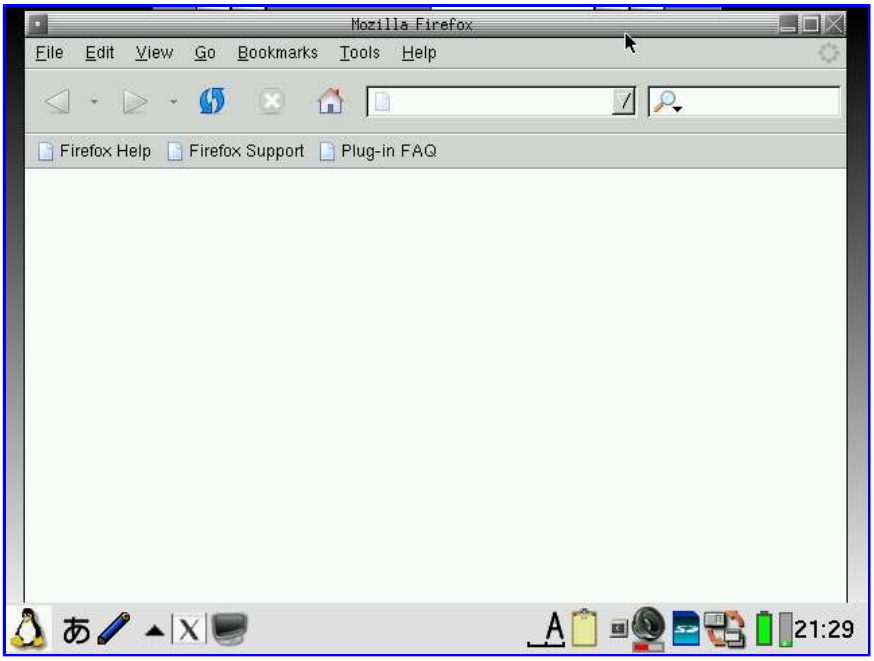
You can also create an icon on the Qt desktop for Firefox (unless you are using the modified version which already includes it). Create a firefox.desktop file in the appropriate location, eg /home/QtPalmtop/apps/Applications that looks like this:

```
[Desktop Entry]
Name = Firefox
Exec = runfirefox
Comment = Mozilla Firefox
Icon = mozicon50
Type = Applications
Display = 640x480/144dpi,480x640/144dpi
```

Now create a file /home/QtPalmtop/bin/runfirefox that looks like this:

```
#!/bin/sh
xlauncher firefox
```

Make runfirefox executable. This causes firefox to be launched and X is only loaded if it is not loaded yet.



Thunderbird also requires X/Qt and works nicely as well if you have the X render update [xqt-libXrender_1.2.2_arm.ipk]. Make sure you install it if you manually installed X/Qt. If you have installed one of the xqt-jumpbopacks then there is no need to install it again since they alread include it.

To install thunderbird you need either the original [thunderbird_0.6_armv5tel.ipk] or the modified [thunderbird_0.6-3_arm.ipk]. If you are using the original package, you need to also install the pdaXrom compatible libraries [libstdc5-compat-sharp_0.5_arm.ipk] and [libiconv_1.8-2_arm.ipk].

On a C3000, space on /home is quite scarce so it is better to move the thunderbird profiles to somewhere else with more space as well.

```
# su
# mkdir -p /hdd3/zaurushome
# mv /home/zaurus/.thunderbird /hdd3/zaurushome
# ln -s /hdd3/zaurushome/.thunderbird /home/zaurus/.thunderbird
```

Mozilla Firefox and Thunderbird on Debian

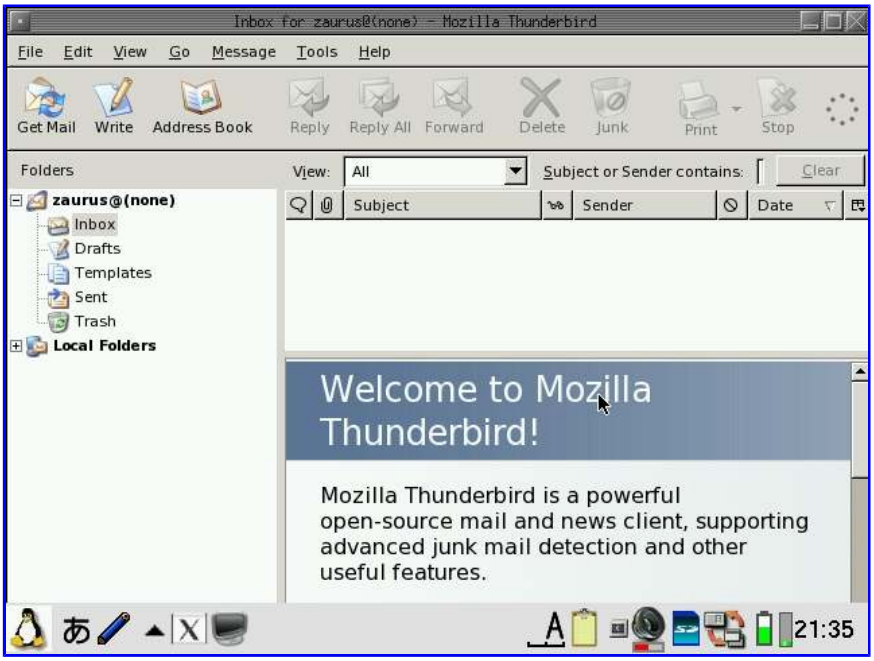
As an alternative, the Debian build of Mozilla Firefox and Thunderbird can also be installed and they work perfectly. You will first need to install PocketWorkstation. Once that is done installing Firefox and Thunderbird is quite easy. From within a terminal in Debian:

```
# source /root/.profile
# apt-get update
# apt-get install mozilla-firefox
# apt-get install mozilla-thunderbird
# apt-get clean
```

This assumes you are connected to the net. apt-get will download all the required packages and then install the mozilla app you selected. The following is a list of files it downloads and installs:

- cpp-3.3_1%3a3.3.5-12_arm.deb
- expat_1.95.8-3_arm.deb
- gcc-3.3-base_1%3a3.3.5-12_arm.deb
- libatk1.0-0_1.8.0-4_arm.deb
- libexpat1_1.95.8-3_arm.deb
- libfontconfig1_2.3.1-2_arm.deb
- libgcc1_1%3a3.4.3-12_arm.deb
- libglib2.0-0_2.6.4-1_arm.deb
- libgtk2.0-0_2.6.4-1_arm.deb
- libgtk2.0-bin_2.6.4-1_arm.deb

- libgtk2.0-common_2.6.4-1_all.deb
- libidl0_0.8.5-1_arm.deb
- libkrb53_1.3.6-2_arm.deb
- libpango1.0-0_1.8.1-1_arm.deb
- libpango1.0-common_1.8.1-1_arm.deb
- libpng12-0_1.2.8rel-1_arm.deb
- libstdc++5_1%3a3.3.5-12_arm.deb
- libtiff4_3.7.2-2_arm.deb
- libxcursor1_1.1.3-1_arm.deb
- mozilla-firefox_1.0.3-2_arm.deb
- mozilla-thunderbird_1.0.2-2_arm.deb



Installing X/Qt applications

In order to use any of the following X/Qt applications, X/Qt needs to be installed first. See the X/Qt and/or X/Qt jumbo sections on how to do that. Once you have X/Qt installed and configured, you can install any of the following applications:

- abiword (Word Processor)
- flfm (File Manager)
- fltdj (Personal Information Manager)
- gimp (Graphics)
- gnumeric (Spreadsheet)
- xmms (Multimedia Player)

Installing QuantumStep

QuantumStep is a MacOSX environment for the Zaurus running under X/Qt. It comes with its own X/Qt binaries which is good for you if you haven't got X/Qt installed yet, but messes up your existing X/Qt setup slightly.

By default, the QuantumStep installer extracts the files to /home/myPDA which is the flash on the C3100. For the C3000, it relinks it to /hdd2/myPDA before extracting the files.

If you want to install it to a different location, you need to have an ext2/ext3 formatted disk either natively, or as a loopback filesystem. You will need at least 20MB of disk space for QuantumStep. If you create a loopback filesystem, make it at least 32MB.

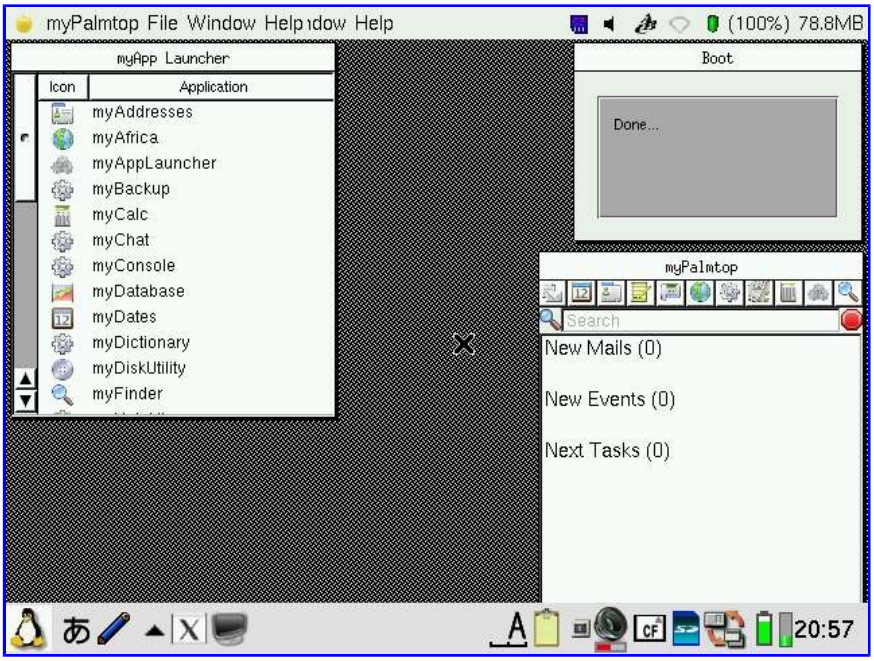
To manually install it, you will need the following files:

- QuantumSTEP-ZED-DR7.mypkg
- quantumstep-setup

You will first need to extract QuantumSTEP-ZED-DR7.mypkg to an ext2/ext3 formatted disk, eg:

```
# su
# dd if=/dev/zero of=/hdd3/quantumstep.ext3
# echo y | mke2fs -j /hdd3/quantumstep.ext3
# mkdir -p /mnt/quantumstep
# echo "/hdd3/quantumstep.ext3 /mnt/quantumstep ext3 loop,rw,noatime 0 0" >> /etc/fstab
# mount -o loop /hdd3/quantumstep.ext3 /mnt/quantumstep
# zcat QuantumSTEP-ZED-DR7.mypkg | tar xvf - -C /mnt/quantumstep
# cp quantumstep-setup /mnt/quantumstep
# chmod 755 /mnt/quantumstep
# echo "echo QuantumStep DR7.0" > /mnt/quantumstep/version
#chmod 755 /mnt/quantumstep/version
# cd /mnt/quantumstep
# ./quantumstep-setup
```

This is how it looks like once installed:



You might want to move the quantum step icon from the Application tab to the X/Qt tab, and you can delete the quantumstep icon from the Settings tab.

Installing Wellenreiter

You need to install the following packages:

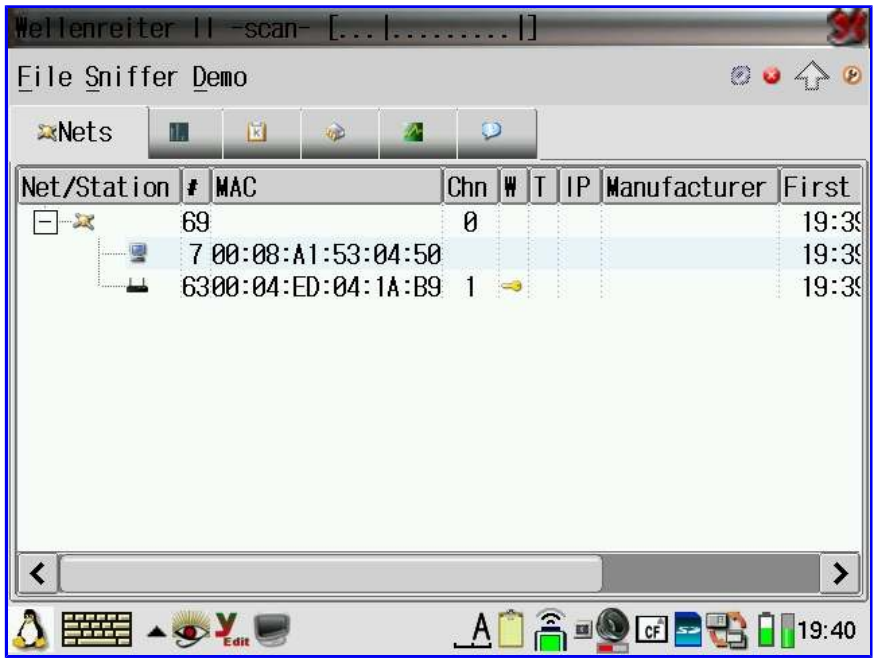
- libcopie1 - [libopie1_1.1.0-20031220_arm.ipk]
- libcopie2 - [libopie2_1.8.2-20031220_arm.ipk]
- libpcap - [libpcap0_0.7.2-20031220_arm.ipk]
- opie-wellenreiter - [opie-wellenreiter_1.0.2.1-20031220_arm.ipk]

Once Wellenreiter is installed, create a network profile for it as follows:

```
Account
  Name: wellenreiter
Config
  Non-Spec ESS-ID: "ANY" (unticked)
  ESS-ID: test
  Network Type: 802.11 Ad-Hoc
WEP
  Key Type: Disabled
PPoE
  Use PPoE Authentication (unticked)
```

```
WEB Auth
  Use WEB Authentication (unticked)
TCP/IP
  Obtain TCP/IP information Automatically (unticked)
  IP Address: 1.1.1.1
  Subnet Mask: 255.255.255.0
  Gateway: 1.1.1.0
DNS
  Auto-detect name servers (ticked)
  Default domain: (leave empty)
Proxy
  No proxy
```

Connect to this network and then start wellenreiter.



Installing Kismet

You need to install the following packages:

- libstdc6 - [libstdc6_1.2.2_arm.ipk]
- libpcap - [libpcap0_0.7.2-20031220_arm.ipk]
- kismet - [kismet_2005.08.R1_arm.ipk] (contained in kismet-2005-08-R1-arm.tar.gz)
- kismet-qt - [kismet-qt_2.0.0_arm.ipk]
- kismet-misc - [kismet-misc_0.3_arm.ipk]
- netctl - [netctl_0.3.0-1_arm.ipk]

Once you have installed the packages, you need to do the following first if you have a C3000 :

```
# su
# mkdir -p /home/root/usr
# ln -s /opt/QtPalmtop/lib /home/root/usr/lib
```

Then you need to symlink libpcap.so.1 as follows

```
# su
# ln -s /home/root/usr/lib/libpcap.so.0.6.2 /home/QtPalmtop/lib/libpcap.so.1
```

Now modify /usr/local/etc/kismet.conf to update the following entries:

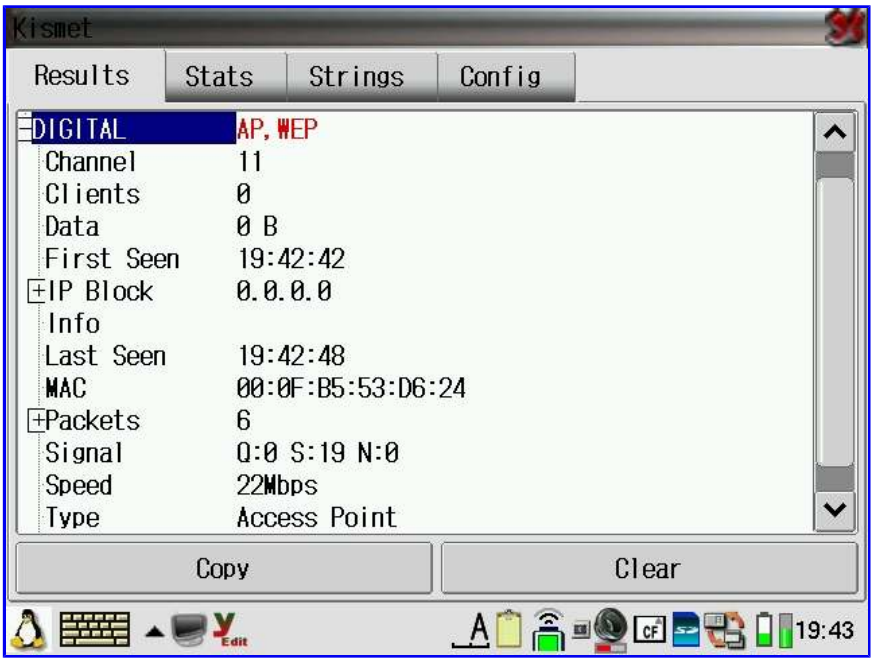
```
suiduser=zaurus
```

```
source=wlanng,wlan0,wireless
sound=true
speech=true
festival/usr/local/bin/flite
flite=true
logtemplate=/home/zaurus/Documents/%n-%d-%i.%l
```

Once that is done create a network profile as follows:

```
Account
  Name: kismet
Config
  Non-Spec ESS-ID: "ANY" (unticked)
  ESS-ID: any
  Network Type: 802.11 Ad-Hoc
WEP
  Key Type: Disabled
PPoE
  Use PPoE Authentication (unticked)
WEB Auth
  Use WEB Authentication (unticked)
TCP/IP
  Obtain TCP/IP information Automatically (unticked)
  IP Address: 10.1.0.2
  Subnet Mask: 255.0.0.0
  Gateway: 10.1.0.1
DNS
  Auto-detect name servers (unticked)
  Primary DNS: 10.1.0.1
  Secondary DNS:
  Default domain: (leave empty)
Proxy
  No proxy
```

Note: you will need to install **flite** for the speech and **sudo** if you want to launch kismet from the GUI



Installing Yahoo Messenger clone Qazoo

This works quite nicely, although I hate the default icon.

You need to install the following packages:

- libyahoo2 - [libyahoo2_cvs-20040713_arm.ipk]
- qazoo - [qazoo_0.8.1_arm.ipk]
- qazoosounds - [qazoosounds_0.8_arm.ipk]

Installing mplayer (and kino2 or zplayer)

This version is specifically written for the C3000 and C3100, and uses the optimised bvdd drivers. Most video formats work, ie. mov, mpg, avi, asf.

You need to install the following packages for the stock Sharp ROM:

- libffmpeg - [libffmpeg_0.4.6_20030304_arm.ipk]
- libiconv - [libiconv_1.8-2_arm.ipk]
- kino2 - [kino2_0.4.3c_arm.ipk]
- bvdd - [bvdd_0.4.0-1_arm.ipk]
- mplayer-bvdd - [mplayer-bvdd_1.1.5-1_arm.ipk]
- zplayer - [zplayer_0.1.0_arm.ipk]

The intel wireless media extension (iwmmxt) is a feature of the Xscale CPUs which improves performance dramatically. It is utilised by the special iwmmxt edition of mplayer. It is faster than the regular version without iwmmxt but you will need Tetsu's kernel installed to take advantage of it since it needs some extra patches which the default Sharp kernel is missing but is included in Tetsu's special kernel.

kino2 and zplayer are graphical frontends for the console based mplayer. You don't need them, but they make things easier. You can use either of them with mplayer or none at all if you prefer to run mplayer from command line.

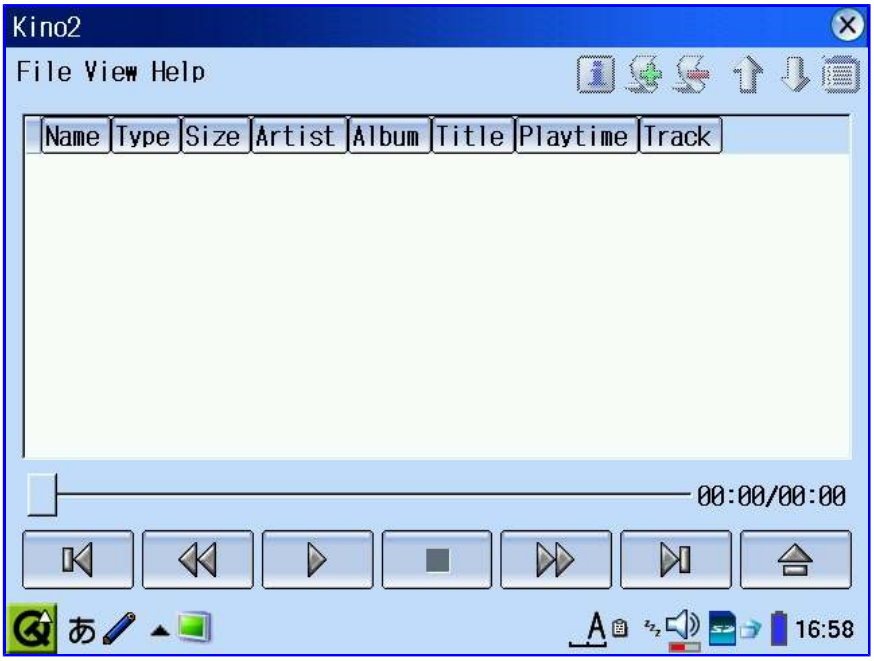
If you want to resize the default video aspect or compress them, have a look further down in the Video Conversion section.

In order to play videos fullscreen in kino2, you need to have the following settings set and enabled:

> Video
>
> - Use PXA27X(bvdd) overlay *(tick)*

This will let you play videos with 320x240 resolution in full screen. However, if you have files encoded with the standard PAL resolution of 352x288 then you need to have the following options as well:

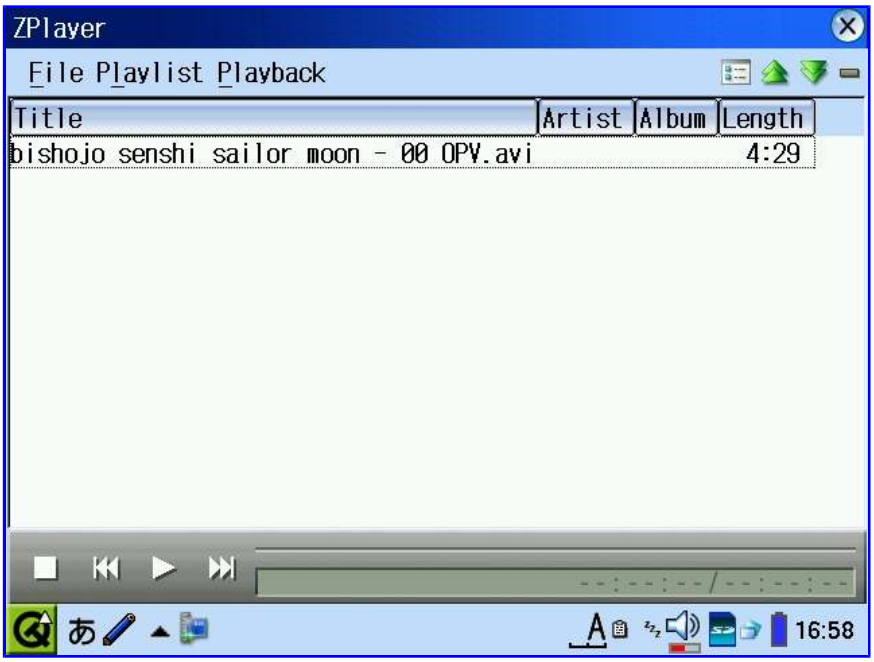
> Video
>
> - Enable smart crop *(tick)*
> - Disable aspect ratio correction *(untick)*



You might also want to enable some of the other settings which may give you better performance. See the Video Conversion section to see how you can re-encode your videos to make them smaller but still give you reasonable playback quality.

zplayer can play videos in window mode or fullscreen if they are encoded with 320x200 resolution without adding extra parameters. However, in order to play 352x288 encoded PAL videos in fullscreen mode, you will need to provide the following mplayer options under Tools->Configuration:

```
-vo bvdd -vm -x 320 -y 200
```



The following mplayer command line will allow you to watch fullscreen video:

```
# mplayer -vo bvdd -afm libmad -vm -vf crop=320:240 file
```

The following mplayer options will give you better and smoother video playback:

```
# mplayer -nortc -noaspect -double -framedrop -cache 2048 -dr -vo bvdd -afm libmad
-af lavcresample=44100 -vf crop=320:240 file
```

And here is a list of the most useful mplayer options:

- -vm
- -vf-add scale=16:9
- -vf crop=320:240, rotate=1
- -af lavcresample=44100:0:0
- -afm libmad
- -hr-mp3-seek
- -dr
- -double
- -framedrop
- -noaspect
- -nortc
- -cache 1024
- -autosync 100

Installing Opera

Although this is an older package, it will still work with the C3000 and C3100.

You need to install the following packages:

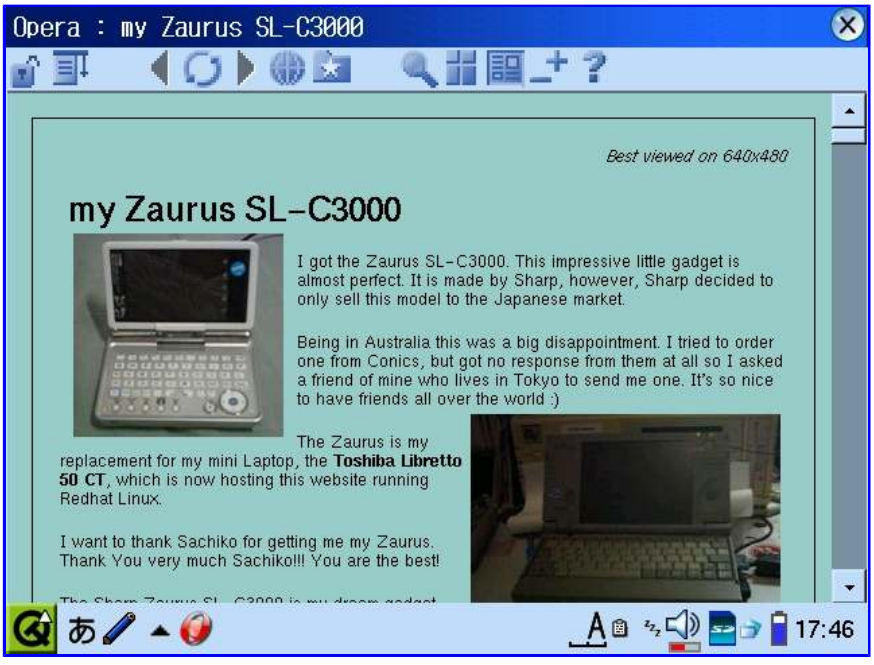
- opera_sl-5x00-7.30.9965_arm.ipk
- opera-cseries-fix_7.30_arm.ipk

Once you installed the packages, you still need to do the following:

```
# su
# ln -s /opt/QtPalmtop/opera /usr/share/opera
# chown -R zaurus /home/zaurus/.opera
# chown -R zaurus /home/zaurus/.operasave

(optionally)

# rm -r /home/QtPalmtop/opera/voice/
# rm -r /home/QtPalmtop/opera/start/
```


Opera : my Zaurus SL-C3000
Best viewed on 640x480
my Zaurus SL-C3000
I got the Zaurus SL-C3000. This impressive little gadget is almost perfect. It is made by Sharp, however, Sharp decided to only sell this model to the Japanese market.
Being in Australia this was a big disappointment. I tried to order one from Conics, but got no response from them at all so I asked a friend of mine who lives in Tokyo to send me one. It's so nice to have friends all over the world :)
The Zaurus is my replacement for my mini Laptop, the Toshiba Libretto 50 CT, which is now hosting this website running Redhat Linux.
I want to thank Sachiko for getting me my Zaurus. Thank You very much Sachiko!!! You are the best!
17:46

Alternatively, some newer packages have been build for the C3000 and C3100. You can try one of these instead:

- opera_7.25_c3k_1_arm.ipk
- opera_7.55.6079-SLC3000_arm.ipk

Installing Doom

To install Doom, you need to install the following packages:

- libsdl - [libsdl_1.2.5-slzaurus20050731_arm.ipk]
- libsdl-mixer - [libsdl-mixer_1.2.5cvs-1_arm.ipk]
- libsdl-net - [libsdl-net_1.2.5cvs-1_arm.ipk]
- doomdemo - [doomdemo_1.8-1_arm.ipk]
- prboom - [prboom_2.2.3-2_arm.ipk]

Once installed, copy prboom.cfg to /home/zaurus/.prboom, this file stores the doom key mappings which need to be fixed for the Zaurus and you can specify additional wad files too. The wad files should go into /home/QtPalmtop/shared/games/doom. You may want to copy doom.wad and doom2.wad and any other wad files you might have into that directory.

Doom runs quite nicely and you can use either doom1 or doom2 wad files.

Installing Quake

To install Quake, install the following packages:

- libsdl - [libsdl_1.2.5-slzaurus20050731_arm.ipk]
- libsdl-mixer - [libsdl-mixer_1.2.5cvs-1_arm.ipk]
- libsdl-net - [libsdl-net_1.2.5cvs-1_arm.ipk]
- quake - [qpe-quake_1.5.0-2_arm.ipk]
- quake-data - [qpe-quake-data_1.5.0-1_arm.ipk]

Change /home/QtPalmtop/bin/run_quake and update the line which looks like this:

quake -nosound -width 150 -height 135 -basedir /opt/QtPalmtop/quake/data

to:

```
quake -fullscreen -width 320 -height 240 -basedir /opt/QtPalmtop/quake/data
```

Quake is awfully slow in fullscreen mode, but even if you reduce the window size, it is still quite slow.

You can also try QuakeII under Debian. This assumes you already have a working PocketWorkstation. To install quake2, you first need to make sure apt-get is able to find it. To do that edit **/etc/apt/sources.list** and add the following:

```
deb http://ftp.debian.org/debian sarge contrib
```

Then do the following:

```
# source /root/.profile
# apt-get update
# apt-get install quake2
# apt-get clean
```

This assumes you are connected to the net. apt-get will download all the required packages and then install quake2. The following is a list of files it downloads and installs:

- libao2_0.8.6-1_arm.deb
- libasound2_1.0.8-3_arm.deb
- libsdl1.2debian-oss_1.2.7+1.2.8cvs20041007-4.1_arm.deb
- libsdl1.2debian_1.2.7+1.2.8cvs20041007-4.1_arm.deb
- quake2-data_13_all.deb
- quake2_1%3a0.3-1.1_arm.deb

It also downloads the demo version of quake2 from Id's site and extracts the pak file. Once installed, you can add to or replace the pak files which are located at /usr/share/games/quake2/baseq2. The quake2 executable is /usr/lib/games/quake2/quake2.real if you prefer to run it directly from the command line instead of launching it from the icewm menu.

Installing Heretic

To install Heretic, you need to install the following packages:

- libsdl - [libsdl_1.2.5-slzaurus20050731_arm.ipk]
- libsdl-mixer - [libsdl-mixer_1.2.5cvs-1_arm.ipk]
- libsdl-net - [libsdl-net_1.2.5cvs-1_arm.ipk]
- heretic-demo - [heretic-demo_1.2-3_arm.ipk]
- heretic-engine - [heretic-engine_1.0.4-2_arm.ipk]

Once installed, leave the option "Display with magnified screen" ticked. It will automatically go full screen and have the right landscape orientation.

Installing ltris and other lgames

This applies to the following games:

- barrage - [barrage_1.0.2_arm.ipk]
- lgeneral - [lgeneral_1.2beta-2_arm.ipk]
- lmarbles - [lmarbles_1.0.7-2_arm.ipk]
- ltris - [ltris_1.0.10-2_arm.ipk]

If you cannot save your results or start the game, do the following:

```
# su
# chown -R zaurus /home/zaurus/.lgames
```

Installing Supertux

Once you have installed supertux you need to change /home/QtPalmtop/bin/supertux.sh (this assumes you already have sudo configured to allow you to mount and unmount, see sudo section):

```
#!/bin/sh -e
BASE=`grep "/QtPalmtop/$" /usr/lib/ipkg/info/supertux.list | tail -n 1`
#sudo mount ${BASE}share/supertux.cramfs ${BASE}share/supertux -o ro,loop
sudo mount ${BASE}share/supertux.cramfs ${BASE}data -o ro,loop
supertux
#sudo umount ${BASE}share/supertux
sudo umount ${BASE}data
```

Installing FreeCiv

You need the following packages/files:

- stdsounds2.tar.gz

- freeciv_zaurus_0_0_5_bin.tar.bz2

To install freeciv do the following:

```
# su
# mkdir -p /home/QtPalmtop/share/freeciv
# bzip2 -dc freeciv_zaurus_0_0_5_bin.tar.bz2 | tar xvf - -C /home/QtPalmtop/share/freeciv
# mv /home/QtPalmtop/share/freeciv/civclient.png /home/QtPalmtop/pics
# zcat stdsounds3.tar.gz | tar xvf - -C /home/QtPalmtop/share/freeciv
# ln -s /home/QtPalmtop/share/freeciv/data/stdsounds.soundspec
/home/QtPalmtop/share/freeciv/data/stdsounds.spec
# vi /home/QtPalmtop/bin/runciv
```

```
#!/bin/sh
CIVHOME=/opt/QtPalmtop/share/freeciv
$CIVHOME/civserver&
bg
$CIVHOME/civclient
kill -9 `ps -ef|grep civ|grep -v grep|awk '{print $2}'`
sleep 5
kill -9 `ps -ef|grep civ|grep -v grep|awk '{print $2}'`
```

```
# chmod 755 /home/QtPalmtop/bin/runciv
```

Create freeciv.desktop in /opt/QtPalmtop/apps/Games as follows:

```
[Desktop Entry]
Comment=Freeciv
Exec=runciv
Icon=civclient
Type=Application
Name=FreeCiv
Display=640x480/144dpi,480x640/144dpi
```

Run the TabSetting config app under Settings and press OK to save the new config. This will result in the freeciv icon to be visible in the Games tab without rebooting or restarting Qtopia.

Installing pico

Pico depends on the ncurses library so in order to install pico, install the following packages in the given order:

- libncurses_5.0_arm.ipk
- pico_4.4_arm.ipk

Installing file

Once you have installed file [file_3.39-2_arm.ipk] it will not work until you do the following:

```
# file -C
```

Installing Cacko packages

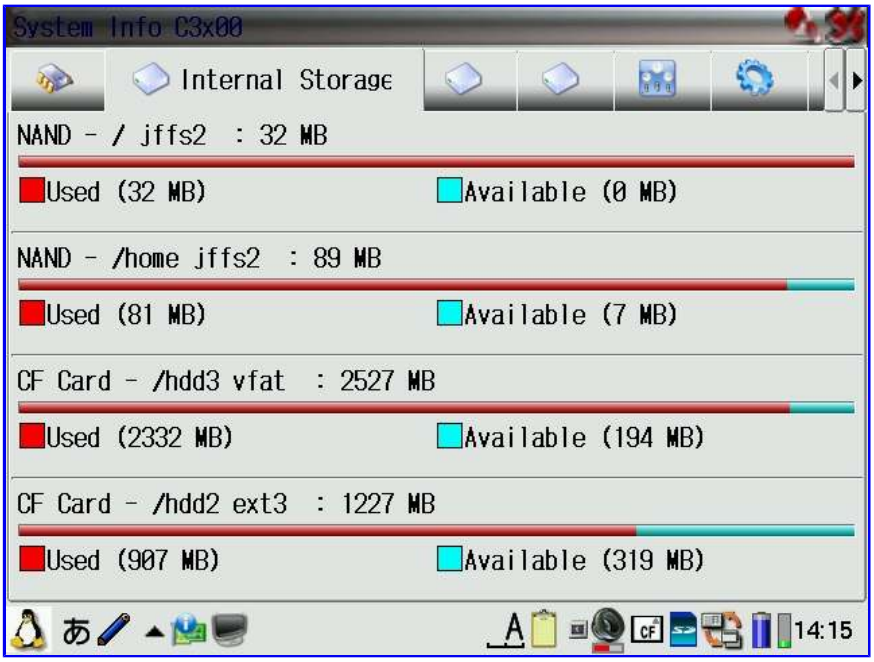
Packages for Cacko and Sharp ROM are generally compatible and interchangable. You can install Cacko packages onto the Sharp ROM and use Cacko feeds.

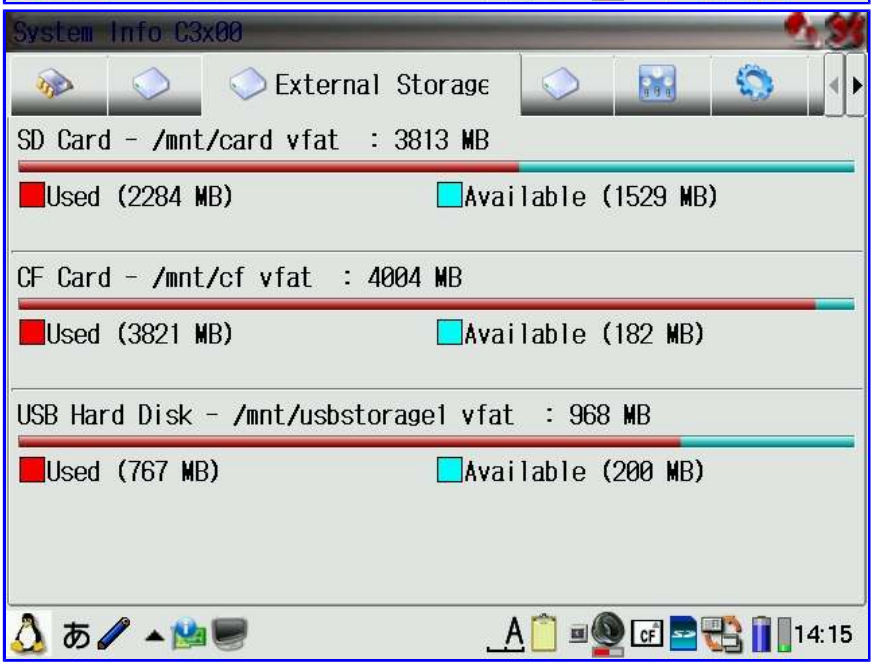
Cacko is basically an enhanced and customised Sharp ROM. As such, there are applications that are pre-installed on Cacko but don't exist on Sharp ROM. See the Cacko sub section under the Alternate Distros/ROMs section for more details about the differences between Cacko ROM and Sharp ROM.

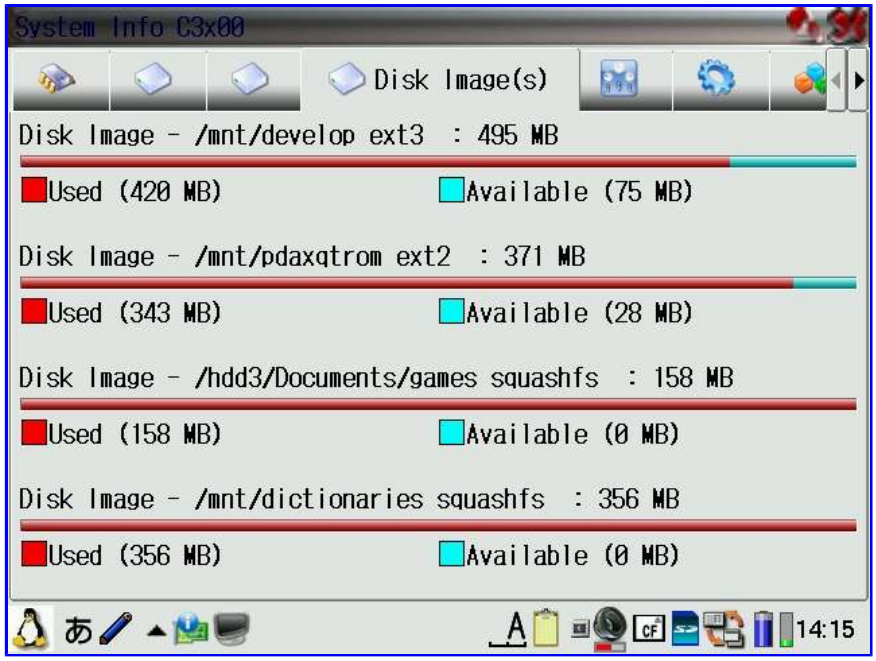
Here are some packages extracted from Cacko that can be installed to enhance Sharp ROM:

- **qtopia-sysinfo_1.23_arm.ipk** - enhanced sysinfo tool with process and mount controls
- **qtopia-addressbook_1.23_arm.ipk** - addressbook with alphanumeric sorting support
- **qtopia-combbatteryapplet_1.0.6_arm.ipk** - updated battery applet with overclocking/underclocking support
- **qtopia-memoryapplet_1.0.3_arm.ipk** - updated memory applet with better swapfile management (there is a more updated version which supports larger swap files)
- **qtopia-keyboardapplet_1.0.0_arm.ipk** - keyboard layout mapper applet
- **qtopia-network-usblan_1.0.0_arm.ipk** - network config for usb lan adaptor
- **qtopia-network-bluetooth_1.0.0_arm.ipk** - network config for bluetooth adaptor
- **vga-console-font_1.0-1_arm.ipk** - vga console font

qtopia-sysinfo_1.23-2_arm.ipk is an even more enhanced sysinfo tool based on the Cacko one and displays more detailed disk info.


System Info C3x00
Internal Storage
NAND - / jffs2 : 32 MB
Used (32 MB)
Available (0 MB)
NAND - /home jffs2 : 89 MB
Used (81 MB)
Available (7 MB)
CF Card - /hdd3 vfat : 2527 MB
Used (2332 MB)
Available (194 MB)
CF Card - /hdd2 ext3 : 1227 MB
Used (907 MB)
Available (319 MB)
14:15


System Info C3x00
External Storage
SD Card - /mnt/card vfat : 3813 MB
Used (2284 MB)
Available (1529 MB)
CF Card - /mnt/cf vfat : 4004 MB
Used (3821 MB)
Available (182 MB)
USB Hard Disk - /mnt/usbstorage1 vfat : 968 MB
Used (767 MB)
Available (200 MB)
14:15


System Info C3x00
Disk Image(s)
Disk Image - /mnt/develop ext3 : 495 MB
Used (420 MB)
Available (75 MB)
Disk Image - /mnt/pdaxqtrom ext2 : 371 MB
Used (343 MB)
Available (28 MB)
Disk Image - /hdd3/Documents/games squashfs : 158 MB
Used (158 MB)
Available (0 MB)
Disk Image - /mnt/dictionaries squashfs : 356 MB
Used (356 MB)
Available (0 MB)
14:15

qtopia-memoryapplet_1.0.4_arm.ipk is a modified version of the memory applet for the C3x00 series that can create a swapfile on /hdd3 and larger swapfile size up to 512MB.

Building your own Packages

You can also build your own packages (ipk files) if you have written some useful scripts or written some applications that you want to distribute and let others install easily with the standard package manager.

I have build a package ipktools [ipktools_0.3.5_arm.ipk] which has a set of tools for manipulating ipk files:

- newipk - creates a package template structure for you to add files to for packaging
- makeipk - package up a directory that contains files in an ipk structure into an ipk file
- unpackipk - extracts the contents of an ipk file into a directory structure for repackaging
- deb2ipk - converts a deb file into an ipk file format
- zipipk - zip up an ipk file and remove ipk file afterwards
- compatipk - attempts to make an ipk packaged for another distro to be more compatible with the Sharp system in terms of filesystem structure
- xipk - installs ipk to alternate location other than main memory
- xipk-link - links files and directory installed to alternate locations (used by xipk)
- xipk-build - used to build cramfs/squashfs images (installs the package to the image without registering it into the package repository)
- ipkg-link - links files and directory installed by ipkg
- ipkg-make-index - generates the Packages file needed for a feed
- pkgsize - reports the installed size of the files making up the package
- ar - extract the .deb/.ipk packages
- mkcramfs - create cramfs image
- mksquashfs - create squashfs image

There currently are two ipk file formats. One uses the tar and gz format wheras the other uses a different format that is similar or the same as the format used for deb files. The Zaurus with default Sharp ROM uses the tar and gz format, which basically is a gzipped tarball (.tgz or .tar.gz) with a control structure and renamed to .ipk. If you extract this ipk file, you will find 3 files inside it - a text file called debian-binaries which just contains the string **2.0**, and two .tar.gz files called control and data. The control.tar.gz file contains a text file called control which has informaton about the package such as the Maintainer's name, dependencies, version, description, etc. There may also be some optional shell scripts for doing some pre and post configuration tasks during install and uninstall. Finally, the file data.tar.gz contains all the files and directory structure of the files for their destination location.

To unpack an ipk file to see what is inside it, do the following:

```
# unpackipk somefile.ipk
```

To create your own ipk file, do the following to create the ipk file structure:

```
# newipk myprojectk
```

Then once you add your files in the correct locations and also update the control file with the information about your application, you can create your ipk file with the following command:

```
# makeipk myprojectk
```

X/Qt Jumbo:

Installing X/Qt ipk file

Installing X/Qt using the jumbo package is relatively simple. It is installed just like any other package, however, due to its size, make sure you have at least 40MB of free disk space (on your destination for the install) to install the xqt jumbo package. Also, temporarily disable suspend during the install since it takes a long time and you don't want your Zaurus to go to sleep in the middle of the install. It is best to install the jumbo package from the command line rather than using the GUI package manager.

The X/Qt jumbo package [xqt-gtk-jumbo_4.3-0.7.1_arm.ipk] installs a complete X server for Qtopia as well as GTK libraries. The X/Qt jumbo package is compatible with the Sharp ROM as well as Cacko, which is a derivative of the Sharp ROM. The X/Qt jumbo package has been tested on the SL-C3100, SL-C3000, SL-C1000 and SL-C860. Other models might work as well, but have not been explicitly tested.

The X/Qt jumbo package is composed of the following packages:

- xqt-fonts-misc [xqt-fonts-misc_4.3.0-3_all.ipk]
- xqt-fonts-100dpi [xqt-fonts-100dpi-iso8859-1_4.3.0-3_all.ipk]
- xqt-fonts-75dpi [xqt-fonts-75dpi-iso8859-1_4.3.0-3_all.ipk]
- xqt-server [xqt-server_1.9.0_arm.ipk]
- xbase-etc [xbase-etc_4.3.0-3_all.ipk]
- zlib - [zlib_1.2.2-1_arm.ipk]
- xlibs [xlibs_4.3.0-3_arm.ipk]
- xbase-client [xbase-clients_4.3.0-3_arm.ipk]
- gdk-pixbuf [gdk-pixbuf_0.22.0-2_arm.ipk]
- glib [glib_1.2.10-0vl3_arm.ipk]
- glib-additional [glib-additional_1.2.10-2_arm.ipk]
- glibc-locale-ja-eucjp [glibc-locale-ja-eucjp_2.2.2-1_arm.ipk]
- glibc-locale-ja-utf8 [glibc-locale-ja-utf8_2.2.2-1_arm.ipk]
- glibc-gconv-ja [glibc-gconv-ja_2.2.2-1_arm.ipk]
- glib2 [glib2_2.4.7_arm.ipk]
- libgcc1 [libgcc1-zaurus_3.2.2-0_arm.ipk]
- freetype [freetype_2.1.5-1_arm.ipk]
- fontconfig [fontconfig_2.2.1-1_arm.ipk]
- fontconfig-etc [fontconfig-etc_2.2.1-1_all.ipk]
- xqt-fonts-encodings [xqt-fonts-encodings_4.3.0-3_all.ipk]
- atk [atk_1.6.1_arm.ipk]
- gtk [gtk_1.2.10.1_arm.ipk]
- gtk2 [gtk2_2.4.13_arm.ipk]
- libtiff [libtiff3.6.1-1_arm.ipk]
- pango [pango_1.4.1_arm.ipk]
- xqtclip [xqtclip_0.0.2_arm.ipk]
- rxvt [rxvt_2.6.4-1_arm.ipk]

- aterm - [aterm_0.4.2_armv5tel.ipk]
- blackbox [blackbox_0.65.0-2_arm.ipk]
- xqt-debian-scripts [xqt-debian-scripts_0.6.1_arm.ipk]

In general, you do not need to re-install the complete xqt-gtk-jumbo package to upgrade it. It takes way too long, but you can do it if you really want to. It is easier and faster to upgrade xqt-gtk-jumbopack by installing a newer version of xqt-debian-scripts or use the xqt-jumbo-upgrade depending on which X/Qt jumbo version is currently installed.

X/Qt jumbopack comes with 0.6.x of xqt-debian-scripts so only install newer versions of xqt-debian-scripts. The main reason to install newer versions of xqt-debian-scripts is to update xlauncher and .xinitrc as well as getting additional X/Qt desktop icons and updated menus.

The X/Qt jumbo package pre-configures all the contained packages so that once it is installed, it is ready to use on the C3x00 and C1000. Other models require the keys to be remapped first. You can either use **xev** to determine the key assigments and create your own .xmodmaprc file or get a pre-made one and place it into /home/zaurus and /home/root (or just symlink them).

To generate the .xmodmaprc file, do the following:

```
# xmodmap -pke > /home/zaurus/.xmodmaprc
```

All you need to do now is use xev to determine the keycodes and update .xmodmaprc to correct the mappings.

The following might get you started:

- xmodmaprc-c3000 - for SL-C3x00 and SL-C1000
- xmodmaprc-c860 - for SL-C8x0 and SL-C7x0
- xmodmaprc-6000 - for SL-6000 and SL-5x00

Once X/Qt is installed, you will get a new **X/Qt tab** and also additional icons on your desktop, however, on some models, you will need to restart Qtopia or reboot your Zaurus in order for them to show up. Use the **StartX** icon to start X-windows. Do not use the **X/Qt Server** icon, but do not remove it either. It is required to display the X/Qt icon on your Qt taskbar when X is running. The other icons are only placeholders and do not do anything until you actually install those applications. Once you install those applications, the icons will be able to launch them. You can also start X from the command line using the **xlauncher** command.

The latest X/Qt jumbo pack would be xqt-gtk-jumbo_4.3-0.7.4_arm.ipk, however, it has not been build as an ipk yet. The xqt-gtk-jumbo cramfs or squashfs image is the equivalent of it.

The default window manager for X/Qt jumbo is now fluxbox which is an enhanced version of blackbox.

The taskbar is at the top of the screen, so it is not hidden behind the Qt taskbar anymore. You can use X control to change X to fullscreen if you don't want to see the Qt taskbar. X control can be activated by pressing the **menu** button which allows you to switch to fullscreen X and/or shutdown X. Pressing **Fn** key + **m** will also shutdown X/Qt. Holding down the **Fn** key while tapping with the stylus will get you the fluxbox control menu.

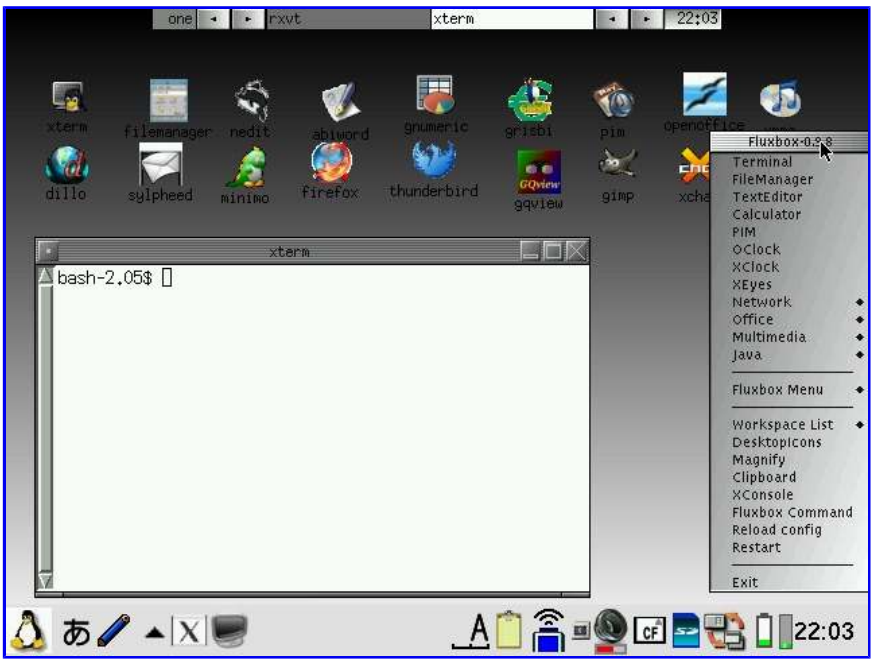
X/Qt jumbo also contains multiple X window managers which you can switch between:

- blackbox
- fluxbox
- evilwm
- twm

To change the default window manager, place the name of the window manager into /etc/X11/defaultwm. For example, to switch to blackbox, do the following and then restart X/Qt:

```
# su
# echo "blackbox" > /etc/X11/defaultwm
```

This is how the custom fluxbox window manager looks like with desktop icons enabled:



However, it takes some time to load and display all the icons, and thus I decided not to load them by default. They really are only eye candy so you need to manually enable them from the fluxbox menu if you want them.

You may find it quicker to start the X/Qt application from the X/Qt tab in the QTopia desktop or using the fluxbox menu. You can also use xlauncher from the console or an exterm to start X applications. Just run xlauncher with the application's executable as the argument and xlauncher will start the application you specified. If X is already running, the application will just appear inside X, but if X has not been started yet, it will start X first before starting the application.

xlauncher will also enable swap if a swapfile exists and is not enabled yet (you really should create a swapfile for running X applications). xlauncher can also be used to allow your X application to be launched from your Qt desktop.

Also, you should install **sudo** if you don't have it installed already (Cacko has it installed by default whereas the default Sharp ROM does not). xlauncher can be run as root if sudo is not available, but it is probably worth it to get sudo installed and configured. Once sudo is properly configured, xlauncher will be able to take advantage of it and be able to enable the swapfile automatically for you.

You can download the xqt-gtk-jumbo package from the following locations:

- [my Zaurus mirror site (http://zaurus.daemons.gr/menaie/mirror/jumbo/xqtipk/)](http://zaurus.daemons.gr/menaie/mirror/jumbo/xqtipk/)
- [Bam's site (http://www.thegrinder.ws/Meanies_XQT/ipk/)](http://www.thegrinder.ws/Meanies_XQT/ipk/)

- Chuckster's site (http://www.chuckster.org/zaurus/)
- Daniel's mirror (http://hplx.mine.nu/daniel/zaurus/xqt)

Updated versions of xqt-debian-scripts can be found at:

- http://www.users.on.net/~hluc/myZaurus/stuff/

The X/Qt jumbo package needs to be installed to main memory. If installing to SD or CF, ipkg-link needs to be run to relink the files and directories, but since ipkg-link does not exist on the standard C3100, C3000 and C1000, I have made a **xipk-link** script (part of my ipktools package) which can relink the X/Qt files and directories so that they can run from SD or CF card.

Installing X/Qt compressed image

I have also created a **xqt-gtk-jumbo.cramfs** and **xqt-gtk-jumbo.squashfs** compressed image which can be installed anywhere you like, since you simply need to mount it and run **xqt-setup** to configure it. If you later decide to move it somewhere else, all you need to do is re-run **xqt-setup** and you are back in business again. This should make installation much simpler and quicker.

I have also created **xqt-install.sh** which will mount the cramfs or squashfs file and run xqt-setup for you. Just place the xqt-gtk-jumbo.cramfs or xqt-jumbo-gtk-jumbo.squashfs file where you want and put the xqt-install.sh script in the same location and do the following:

```
# su
# ./xqt-install.sh
```

If you had installed X/Qt components before and they did not uninstall cleanly, then you can run **xqt-cleanup** before running xqt-install.sh or xqt-setup.

Note: you will need to remount the cramfs image after a reboot and/or re-insert of SD/CF card. If you install my automounter package [automounter-c3000_0.4.7_arm.ipk], then this is automatically taken care of by the script. automounter has only been thoroughly tested for the C3000 and C3100 on the Sharp ROM. Your milage may vary with other variations.

You can download the xqt-gtk-jumbo image from the following locations:

- my Zaurus mirror site (http://zaurus.daemons.gr/menaie/mirror/jumbo/images/)
- Bam's site (http://www.thegrinder.ws/Meanies_XQT/cram/)

The xqt-gtk-jumbo is reported to work for the C3x00, C1000, C8x0, C7x0, 6000 and 5x00. You will need a .xmodmaprc keyboard map listed above for your model.

X applications

There are several X/Qt applications that you can install and use once you have X/Qt. There are some on the X/Qt feeds, and some pdaXrom applications such as Firefox will also work. There is a xqt branch of pdaXrom 1.1 compiled for xqt. Also, the source packages for pdaXrom are also available so someone with time on their hands can always compile additional X/Qt applications. In addition, you can also install Debian Pocketworkstation with X/Qt. Below are tested applications that can be installed for X/Qt. Also the following applications have icons preconfigured for them on the X/Qt tab. You can also create your own launch icons for X/Qt applications. To do that, do the following:

Get Icon

Qt Desktop Icons are usually png files. You can create your own or download them off the net. If you are lucky, the X app might even have installed one under /usr/share/pixmap

Copy the icon file to /opt/QtPalmtop/pics144

Create launch script

Create an executable script that looks like this and place it under /opt/QtPalmtop/bin

```
#!/bin/sh
APPNAME=myapp
if [ "`which $APPNAME`" != "" ]; then
    xlauncher $APPNAME
else
    qcop QPE/TaskBar "message(QString)" "$APPNAME not installed"
fi
```

Call it runmyapp or something like that.

Create desktop file

Create a file such as myapp.desktop that looks like the following and place it under /opt/QtPalmtop/apps/XQt:

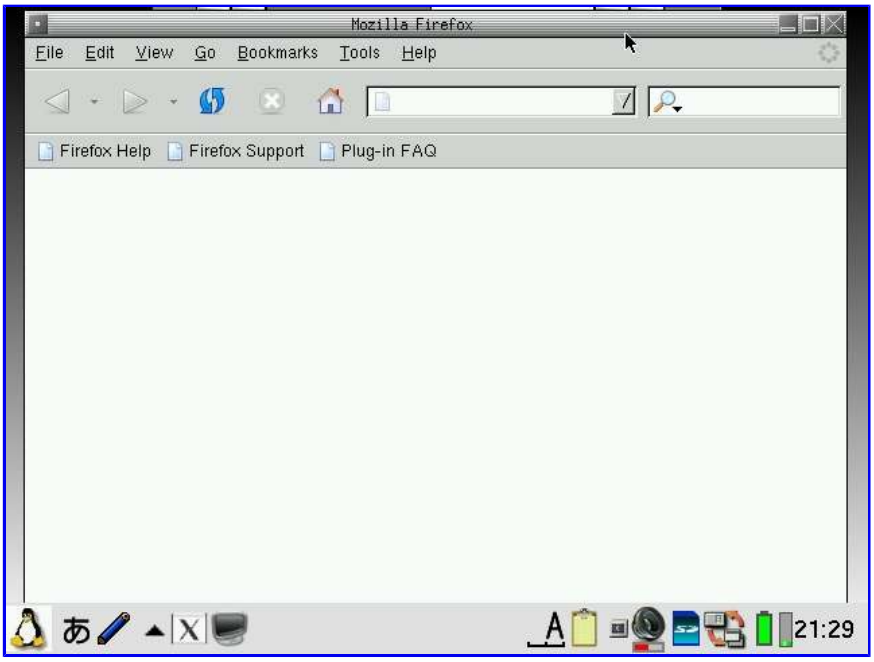
```
[Desktop Entry]
Name = MyApp
Icon = myapp.png
Exec = runmyapp
Comment = My X application
Type = Application
HidePrivilege = 1
Display = 640x480/144dpi,480x640/144dpi
```

Installing Mozilla Firefox

The Mozilla Firefox package from pdaXrom [firefox_0.9gtk_armv5tel.ipk] works with X/Qt on the C3x00. However, other models might have problems running it so I have build a firefox package [firefox_0.9-3_arm.ipk] which hopefully is more compatible and can be run on other models as well as the C3x00.

You can start firefox by either tapping on the Firefox icon on the Qt desktop or launching it from the command line as follows:

```
# xlauncher firefox
```

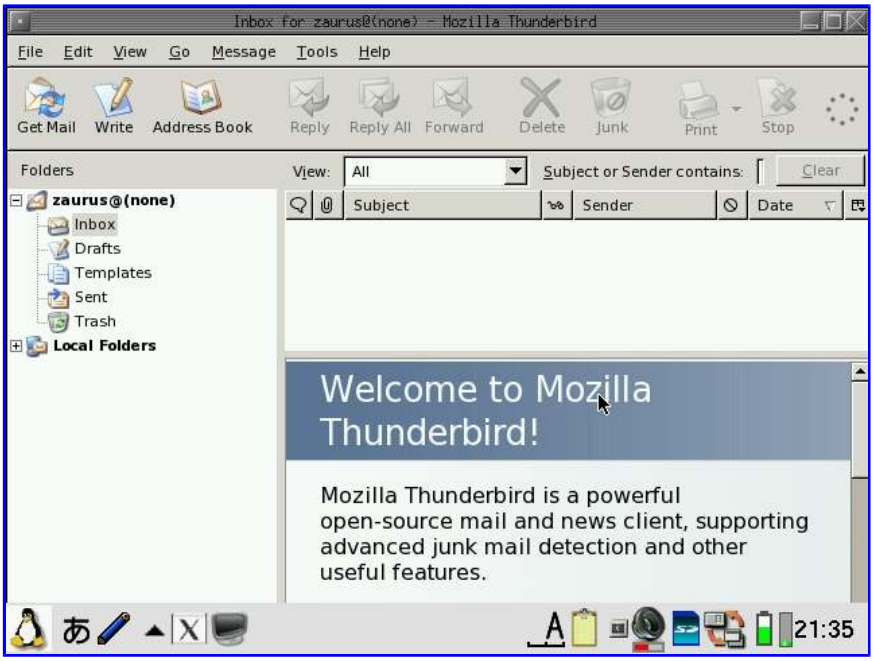


Installing Mozilla Thunderbird

The Mozilla Thunderbird package from pdaXrom [thunderbird_0.6_armv5tel.ipk] works with X/Qt on the C3x00 if you also install libstdc5-compat-sharp [libstdc5-compat-sharp_0.5_arm.ipk]. I have build a thunderbird package [thunderbird_0.6-3_arm.ipk] which includes libstdc5-compat-sharp and it should also be more compatible with other models as well as the C3x00.

You can start thunderbird by tapping on the thunderbird icon or launching it from the command line as follows:

```
# xlauncher thunderbird
```



Installing AbiWord

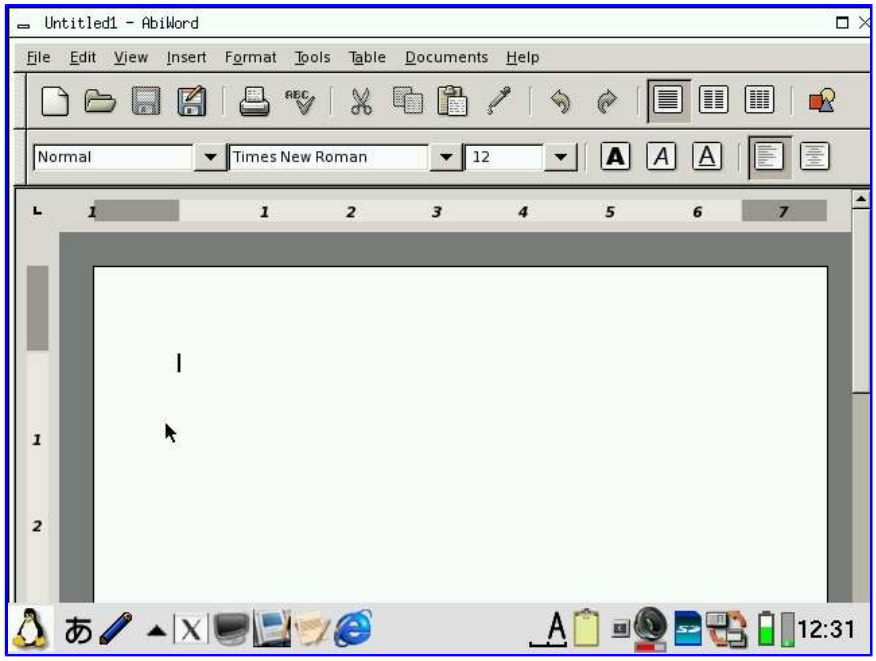
The AbiWord package from pdaXrom works with X/Qt. You will need to install the following packages for AbiWord:

- libiconv - [libiconv_1.8-2_arm.ipk]
- libxml2 - [libxml2_2.6.14-1_arm.ipk]
- libglade - [libglade_2.0.1-1_armv5tel.ipk]
- abiword - [abiword_2.0.0_armv5tel.ipk]

Alternatively, I have build an abiword package [abiword_2.0.0-2_arm.ipk] which includes the above packages.

You can start abiword by tapping on the abiword icon or launching it from the command line as follows:

```
# xlauncher abiword
```



Installing FlFm

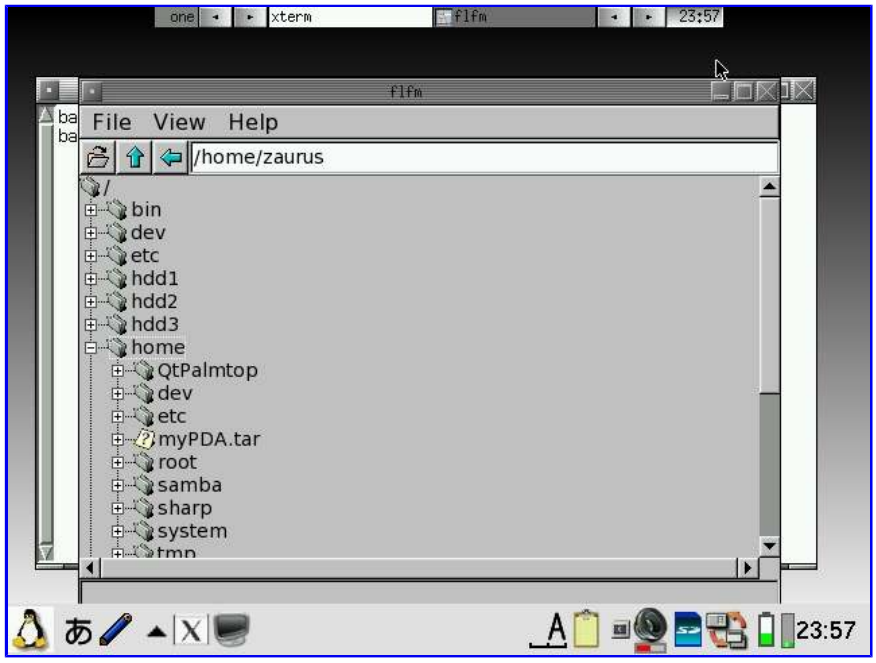
The fl file manager package from pdaXrom works with X/Qt. You will need to install the following packages for flfm:

- flfm_0.3.0_armv5tel.ipk
- xutf8_0.1.1_armv5tel.ipk
- libxd640_0.3.0_armv5tel.ipk

Alternatively, I have build a flfm package [flfm_0.3.0-1_arm.ipk] which includes the above packages.

Once installed you can launch flfm as follows:

```
# xlauncher flfm
```



Installing Fltdj

The fltdj PIM package from pdaXrom works with X/Qt. You will need to install the following packages for fltdj:

- cal_3.5_armv5tel.ipk
- libfltk_1.1.4_armv5tel.ipk
- libfltk-utf8_1.1.4_armv5tel.ipk
- fltdj_0.7_armv5tel.ipk
- fltdj-utf8_0.7_armv5tel.ipk

Alternatively, I have build a fltdj pim package [fltdj_0.7-1_arm.ipk] which includes the above packages.

Once installed you can launch fltdj as follows:

```
# xlauncher fltdj
```

Installing Gimp

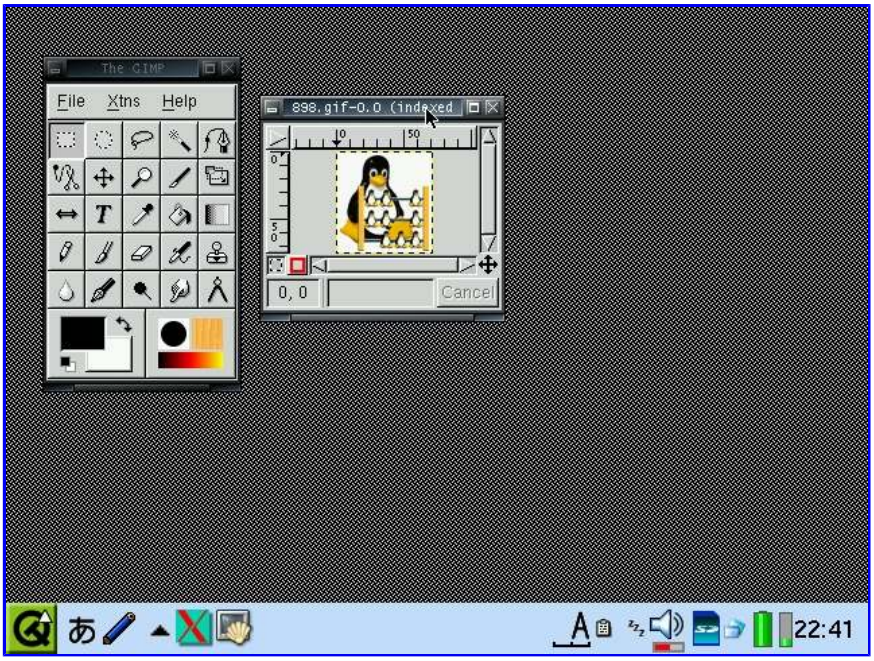
The Gimp and related packages from pdaXrom also work under X/Qt. However, you will need to install several packages to get Gimp working. The following packages are required:

- libgimp - [libgimp_1.2.5-2_armv5tel.ipk]
- xqt-gimp - [xqt-gimp_1.2.5-2_armv5tel.ipk]
- xqt-gimp-plugins - [xqt-gimp-plugins_1.2.5-1_armv5tel.ipk]

libgimp might give some errors, but just ignore them. Once you have installed gimp, you can start it by tapping on the Gimp icon or issuing the following command:

```
# xlauncher gimp
```

I have also created a single gimp package [xqt-gimp_1.2.5-3_arm.ipk] for easier installation, which combines all the above listed packages.



Installing Gnumeric

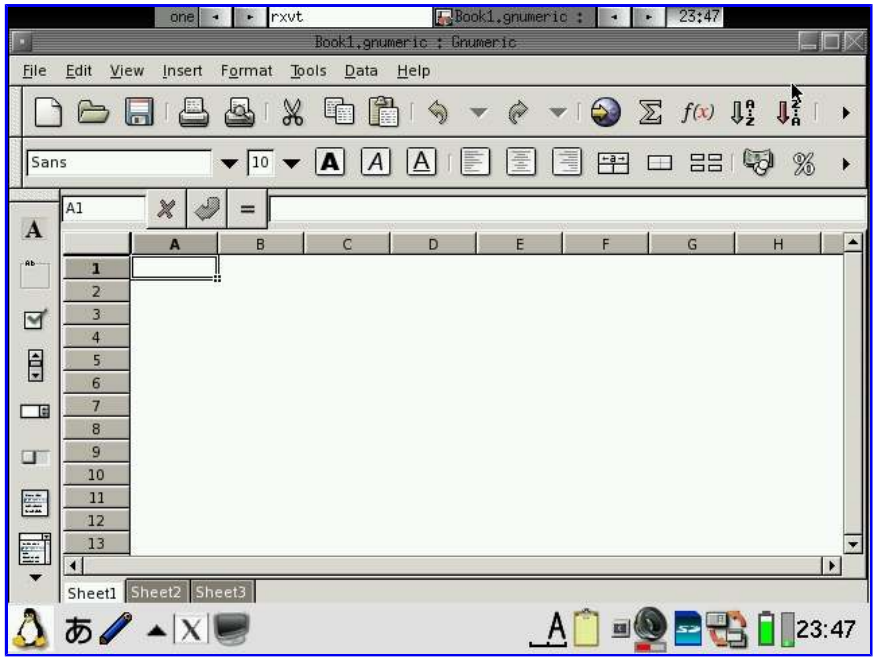
The gnumeric and related packages from pdaXrom also work under X/Qt. However, you will need to install several packages to get gnumeric working. The following packages are required:

- bzip2_1.0.2-1_armv5tel.ipk
- expat_1.95.7-1_armv5tel.ipk
- gconf_2.4.0.1-1_armv5tel.ipk
- gnome-base-libs_2.4.0-1_armv5tel.ipk
- gnome-vfs_2.5.3-1_armv5tel.ipk
- gnumeric_1.2.2-1_armv5tel.ipk
- libart-lgpl_2.3.16-1_armv5tel.ipk
- libbonobo_2.4.2-1_armv5tel.ipk
- libbonoboui_2.4.2-1_armv5tel.ipk
- libglade_2.0.1-1_armv5tel.ipk
- libgnome_2.4.0-1_armv5tel.ipk
- libgnomecanvas_2.5.1-1_armv5tel.ipk
- libgnomeprint_2.4.2-1_armv5tel.ipk
- libgnomeprintui_2.4.2-1_armv5tel.ipk
- libgnomeui_2.4.0.1-1_armv5tel.ipk
- libgsf_1.8.2-1_armv5tel.ipk
- libpopt_1.0.0_armv5tel.ipk
- orbit2_2.9.2-1_armv5tel.ipk

Alternatively, I have build a gnumeric package [gnumeric_1.2.2-1_arm.ipk] which includes the above packages.

Once installed you can launch gnumeric as follows:

```
# xlauncher gnumeric
```



Installing Xmms

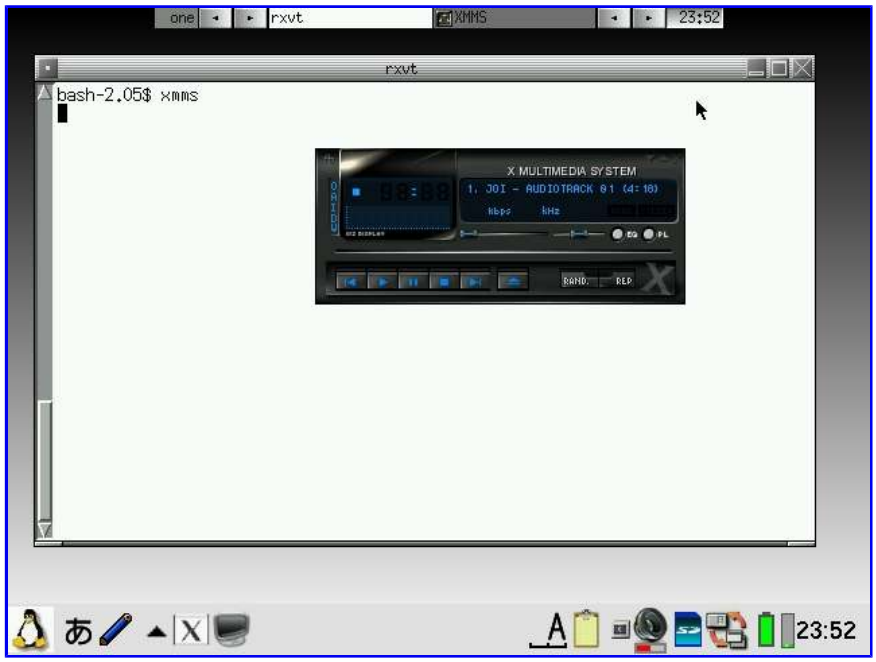
The xmms package from pdaXrom works with X/Qt. You will need to install the following packages for xmms:

- libmad_0.15.1b-1_arm.ipk
- mad_0.14.2b_arm.ipk
- xmms-mad_0.6-1_arm.ipk
- xmms-tremor_1.0_armv5tel.ipk
- xmms_1.2.10j-20040415-1_arm.ipk
- xmmsmplayer_0.4_armv5tel.ipk

Alternatively, I have build a xmms package [xmms_1.2.10-1_arm.ipk] which includes the above packages.

Once installed you can launch xmms as follows:

```
# xlauncher xmms
```



Installing X/Qt applications on cramfs or squashfs

I have also created a **xqt-apps** and a **xqt-mozilla** compressed image in cramfs and squashfs formats which contains the above applications and some extra ones. Installation is the same as for the xqt-gtk-jumbo image. Simply mount the cramfs or squashfs image and then run **xqtapps-setup** to configure the applications.

Currently, the xqt-apps image contains the following applications:

- abiword 2.0
- bluefish 0.13
- bochs 2.1.1
- dia 0.92
- dillo 0.8
- dv 0.0.1
- ethereal 0.10.2
- feh 1.3.4
- flfm 0.3
- fltdj 0.7
- gimp 1.2.5
- gqview 1.4.2
- gnumeric 1.2.2
- grisbi 0.5
- minimo 1.6
- nedit 5.4
- sylpheed 0.9.99
- vncviewer 4.0
- xchat 2.0.5
- xmms 1.2.10
- xpdf 2.03

And the xqt-mozilla image contains:

- firefox 0.9
- thunderbird 0.6

There is also a xqt-openoffice image which runs under X/Qt without requiring PocketWorkstation.

You can also use **xqt-install.sh** just like for xqt-gtk-jumbo. It will also mount the cramfs and/or squashfs file and run xqtapps-setup for you. Just place all the cramfs/squashfs files you want to use into a directory of your choosing and put the xqt-install.sh script in the same location and do the following:

```
# su
# ./xqt-install.sh
```

Note: you can put both xqt-gtk-jumbo and xqt-apps (plus any additional ones) in the same directory as xqt-install.sh and just run it once, however, you can only have one format for each file, either cramfs or squashfs but not both. It will setup and configure all the cramfs or squashfs images. Also remember that you will need to remount the cramfs/squashfs images after a reboot and/or re-insert of SD/CF card. The internal MicroDrive would be a great place to put your cramfs/squashfs images if your Zaurus has one of those since it doesn't need to be ejected. If you install automounter [automounter-c3000_0.4.7_arm.ipk] then the cramfs/squashfs images will be automatically remounted for you on a reboot and mounted/unmounted upon SD/CF card ejection/insertion. The automounter package has only been tested on the Sharp ROM for C3x00 and C1000 and might not work on other models.

You can download the xqt-apps and xqt-mozilla images from the following location:

- my Zaurus mirror site (http://zaurus.daemons.gr/menaie/mirror/jumbo/images/)
- Bam's site (http://www.the grinder.ws/Meanies_XQT/cram/)

Additionally, I have created a java compressed image which contains the full Blackdown JRE 1.3.1 plus a full set of compiler tools making it a JDK in essence (see Java section). It also contains some of my Java applications (Hd Pad and Hd Crawler). Similarly, I have also created a compressed image for an on-board gcc compiler (see gcc section) so you can create your own apps on the Zaurus directly.

You can also create your own custom cramfs or squashfs images with only the applications that you like to have. This involves firstly installing all your desired applications to an ext2 or ext3 formatted partition or disk. You could also install to a fat partition as well but it will fail for some applications so it is safer to

just use an ext2 or ext3 partition. For that, you could reformat your card or create a loopback ext2 or ext3 filesystem.

Once you got your install location prepared, you can use **xipk-build** (found in my ipktools package) to install ipk packages to alternate locations. xipkg by default installs to /hdd3/programs if you have an internal MicroDrive or /mnt/card otherwise. If you want to change the location to something else, then do the following:

```
# su
# echo "/mnt/card/xqt" > /etc/xipk.conf
```

Substitute **/mnt/card/xqt** with your desired install location.

Then install your chosen ipk package as follows:

```
# su
# xipk-build somexqtpackage.ipk
```

Your package is now installed. You can install additional packages if you want to. Once you are finished with the installs, you can compress it as a cramfs image as follows assuming you installed to /mnt/card/xqt and have installed mkcramfs:

```
# su
# cd /mnt/card
# cp `which xipk-link` /mnt/card/xqt/xqt-setup
# echo "echo myXqt apps version 1.0" > /mnt/card/xqt/version
# chmod 755 /mnt/card/xqt/version
# mkcramfs xqt mypackage.cramfs
```

Once the cramfs image is created you can remove, your loopback file you used to build the cramfs image and then mount your new cramfs image as a loopback filesystem and install it using xqt-setup as follows:

```
# su
# mkdir -p /mnt/myxqtapps
# mount -o loop mypackage.cramfs /mnt/myxqtapps
# /mnt/myxqtapps/xqt-setup
```

To create a squashfs image instead, use mksquashfs instead of mkcramfs.

Installing Debian/PocketWorkstation

Debian runs with X/Qt so if you already have X/Qt installed, then you are ready to install Debian. However, if you have not got X/Qt installed yet, then you need to do so before installing Debian. If you just want to run Debian and applications that are part of Debian, but not standalone X/Qt applications then you can also install the Debian jumbo lite package instead of the full xqt jumbo package. Since the xqt-gtk-jumbopack is quite large, I have created the xqt-debian-jumbo-lite package as an alternate package, which only installs the X/Qt libraries required for Debian PocketWorkstation but nothing else. It is much smaller and only requires 11MB, so it does not have gtk or other libraries required to run any of the X/Qt applications mentioned above. You will still be able to run the Debian versions of those applications from within PocketWorkstation, just not directly from X/Qt.

Debian itself is too big to be installed with an ipk file. You can either install Debian using the instructions on my Zaurus website, or install my xqt-debian image file which comes pre-installed with Firefox and Thunderbird, and optionally OpenOffice with the add-on debian-openoffice image file. The packages have been optimised in size so that it is possible to install everything on an ext2 formatted SD or CF card of 512MB. The debian docs are in a separate tarball so you can leave them out if you want to save space. OpenOffice binaries are compressed using a cramfs file image so that everything can fit onto 512MB.

Before installing Debian, make sure you have installed X/Qt yourself or installed one of the following:

- xqt-gtk-jumbo_4.3-0.7_arm.ipk or later
- xqt-debian-jumbo-lite_0.4.3-0.1_arm.ipk
- xqt-gtk-jumbo.cramfs

To install Debian using the image file, unzip xqt-debian-install.zip and place it into the same directory as the other files. The following files are available:

- xqt-debian-install.zip (required)
- zaurus-debian-jumbo-v18-b01.tar.gz (required)
- zaurus-debian-doc-v18.tar.gz (optional)
- zaurus-debian-openoffice114.tar.gz (optional)

The default install location (which has been extensively tested) is /hdd3/debroot. You can specify a different location for the debian root. The install script will by default create a swapfile and pocketworkstation loop image under the same location as your *debroot*, but you can change the location by editing the install script and changing the default locations. If you already have a swapfile located somewhere else, simply comment out the SWAPFILE definition before you install to prevent it from creating an additional swapfile. The script can detect the filesystem type that Debian is being installed to. If the destination filesystem is FAT, it will create an ext2 loopback filesystem for you, but if it is already formatted as ext2 or ext3, then it will simply install there without creating the loopback filesystem.

```
### default settings section start
# swap filename
SWAPFILE=$DEBBASE/swapfile
# swap size (in MB - 64-128)
SWAPSIZE=64
# debian diskimage file
DEBIMG=$DEBBASE/pocketworkstation
# debian size (in MB - 300 or larger)
DEBSIZE=512
# debian tarball
DEBTARBALL=zaurus-debian-jumbo-v18-b01.tar.gz
# debian documentation tarball
DOCTARBALL=zaurus-debian-doc-v18.tar.gz
# openoffice tarball
OOTARBALL=zaurus-debian-openoffice114.tar.gz
### default settings section end
```

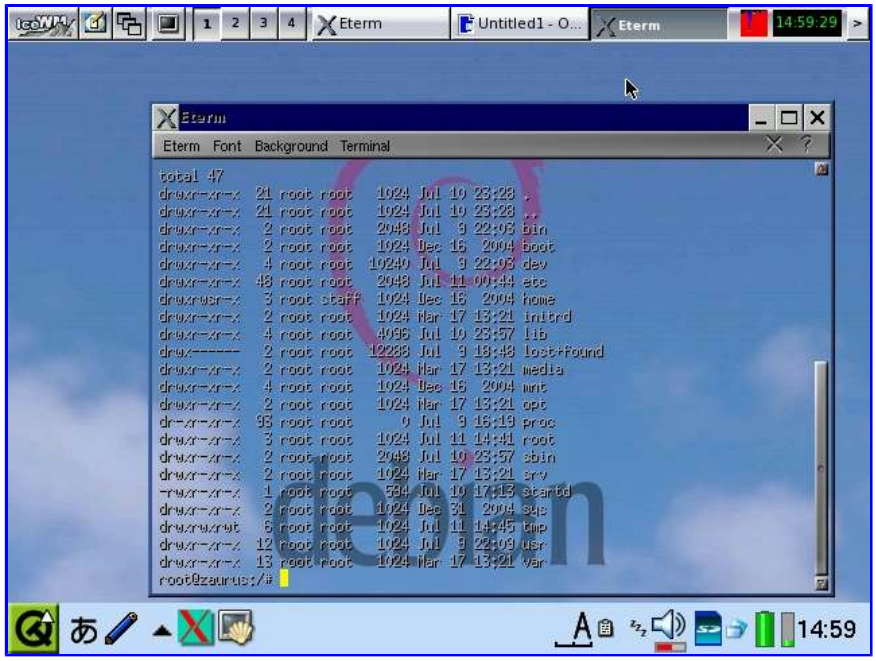
You can increase the swap and pocketworkstation file sizes before you run the install script, but it is not advised to decrease them too much.

From a command line console, do the following:

```
# su
# ./xqt-debian-install.sh /hdd3/debroot
```

Once installed, you can start Debian by tapping on the Debian icon on the Qt desktop or use the command line:

```
# xlauncher debian
```



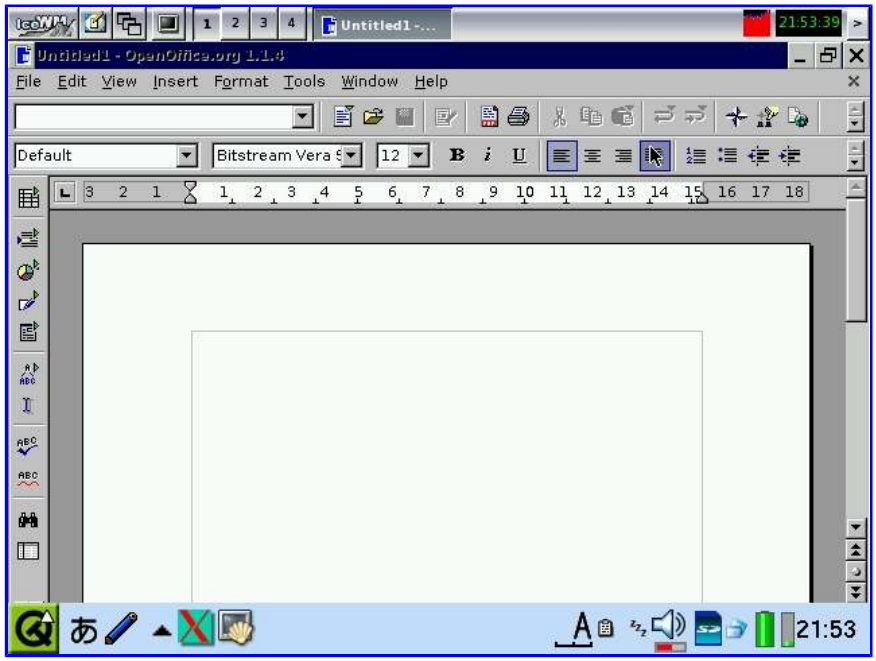
You will need to install **sudo** (refer to customisation section) if you want to run Debian as the zaurus user or launch it via the Qt desktop icon.

The mouse function is emulated as follows:

- Tapping on screen = Left Click
- Fn + Shift + Tapping = Center Click
- Fn + Tapping = Right Click

If you also extracted the openoffice image file, then OpenOffice will be available in Debian. You can start OpenOffice as follows or launch it from the icewm menu:

```
# soffice &
```



If you have space, you replace the openoffice cram file with its extracted version by extracting its content and then removing it. OpenOffice will load faster this way. For best performance, install PocketWorkstation and the swapfile to an ext2 or ext3 formatted CF card and extract the OpenOffice cram file. OpenOffice installed as a cram image with PocketWorkstation on a loopback filesystem and swapfile on the internal microdrive takes about 5 minutes to start. OpenOffice extracted to an ext2 formatted CF card with swapfile located there too takes about 2 minutes to start.

You can download the Debian install files from the following locations:

- my Zaurus mirror site (http://zaurus.daemons.gr/menaie/mirror/jumbo/debian/)
- Bam's site (http://www.thegrinder.ws/Meanies_XQT/deb/)
- Chuckster's site (http://www.chuckster.org/zaurus/)
- Daniel's mirror (http://hplx.mine.nu/daniel/zaurus/xqt)

Note: You can add the optional components even after you have installed Debian. Simply place the optional package in the same directory as xqt-debian-install.sh and re-run it again with the same parameters you had used in the previous install.

Fixes to the installation are provided with newer version of the install script. You can check the version of the install script by doing the following:

```
# grep version xqt-debian-install.sh | head -1
```

Updated debian install script can be found here (version 0.3.8).

Alternatively, you can also use the xqt-debian-install.sh script with zaurus-debian-big-v0.18.tgz to install PocketWorkstation which will also work but you will need to install Firefox, Thunderbird, OpenOffice and Eterm yourself, and you will also only get the default icewm menu and themes.

Running Debian packages without PocketWorkstation

Debian PocketWorkstation requires a lot of space because it is a complete Debian system. You might not want the whole system, but instead you just want to install a few of the Debian packages (.deb files). This is certainly achievable but not as easy as using apt-get to install new Debian packages.

The **deb2ipk** tool which is a Perl script that comes with the ipktools package [ipktools_0.3.5_arm.ipk] converts Debian packages into IPK packages that can be installed onto the Sharp ROM. Most of these Debian packages require X and use a newer standard C library than the sharp ROM. You will need to install X/Qt (the full xqt-jumbo package - [xqt-gtk-jumbo_4.3-0.7_arm.ipk] or later, or the xqt-gtk-jumbo.cramfs, and also an updated C library. This can be achieved with flashing the Zaurus with a zImage that contains libc6 with glib 2.3.2 libraries. Once that is done, it will enable your Sharp ROM to be able to run converted Debian packages without the need for running PocketWorkstation.

The binaries contained in the converted Debian packages should work on the Sharp ROM with the updated C libraries and X/Qt. However, the control scripts might need some fixing before the package can install and run successfully. The **unpackipk** tool can be used to extract ipk package so that the scripts can be manually fixed. Once that is done, the **makeipk** tool can be used to repackage the ipk file (see Building your own Packages section for more details).

If the Debian package has dependencies on other packages, then those also need to be converted and installed as well.

I have already converted and repackaged the following packages from .deb files into .ipk files and tested them to work successfully on the Sharp ROM without PocketWorkstation:

- xqt-icewm_1.2.20-1_arm.ipk - replaces blackbox with icewm using winxp theme
- mozilla-firefox_1.0.4-2_arm.ipk - newer version of firefox

More to come ...

OpenOffice can also be installed directly onto the Sharp ROM using the following with the above mentioned packages installed:

- zaurus-debian-openoffice114.tar.gz
- xqt-openoffice-install.sh (archived as a zip file)

However, getting the above setup is still a bit tricky and risky since you can brick your Zaurus in the process.

OpenOffice as a cramfs or squashfs image

As an alternative, you can run OpenOffice from a cramfs or squashfs image similar to the X/Qt application images. I have bundled OpenOffice with a shrunk down version of Debian (minideb) to allow you to run OpenOffice just like the other X/Qt applications without using PocketWorkstation but utilising minideb as the underlying engine instead.

Installation of xqt-openoffice is similar to the installation of the other xqt cramfs/squashfs images. Simply place xqt-openoffice image into the same directory as xqt-gtk-jumbo image and run xqt-install.sh (make sure to use the latest version of xqt-install.sh)

Once the openoffice cramfs/squashfs image is mounted, you can launch it from the openoffice icon on the X/Qt tab. The openoffice icon with the X/Qt overlayed across it is the icon for the xqt-openoffice version. You can also launch it from the command line as follows:

```
# debrtd -x /opt/OpenOffice/soffice
```

debrtd is the minideb launcher. The -x option tells it to start X if its not already running, and the last argument is the full path to the openoffice launcher on the cramfs (since soffice is not in the path).

---

pdaXQtrom 0.8.2

The new X/QT super jumbo package is the next generation of X/Qt and is based on existing base X/Qt packages (thanks Takuya Murakami) and the specially compiled X/Qt packages from pdaXrom 1.1.0 beta1 (thanks Sashz!). The goal of this image is to extend Sharp ROM with a mini pdaXrom system running under X/Qt. It has a complete X/Qt runtime that is compatible with packages from the special pdaXrom feed. It also uses the same glib2 and gtk+ library versions as pdaXrom while maintaining compatability with Sharp's older glibc 2.2.2 version. This means that most applications available for pdaXrom will be able to be recompiled and run on pdaXqtrom.

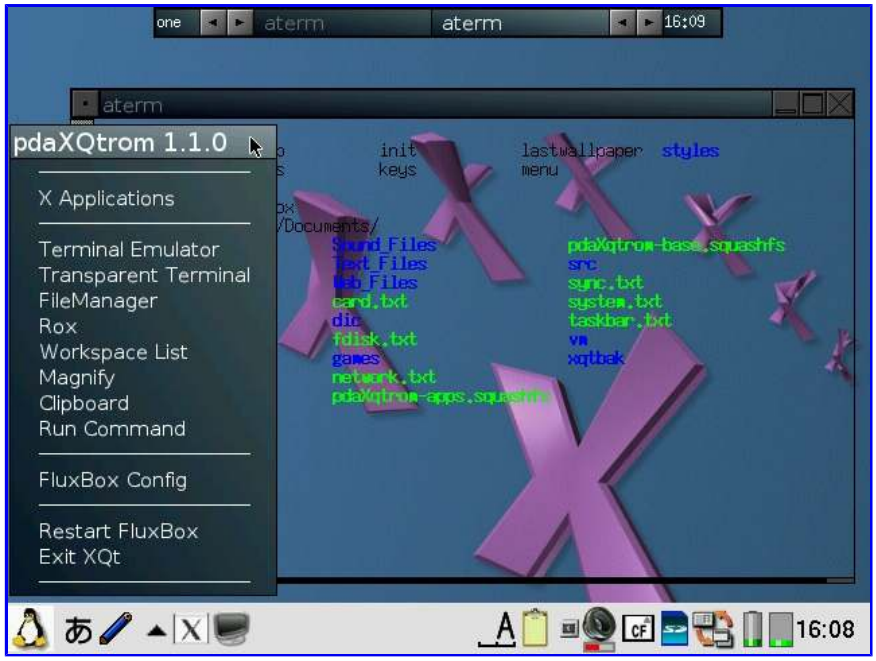
A native on-board compiler based on gcc 2.95.3 is also available as a companion tool to compile your own X/Qt applications from your Zaurus without needing a separate machine or setting up a cross tool chain. A java runtime and development environment is also available as an optional add-on as well.

However, since pdaXqtrom extends Sharp ROM, and many of its libraries are compiled for compatability rather than speed, it is rather slow in comparison to pdaXrom which is optimised for speed. pdaXqtrom is also limited to the constraints of the old version of Qtopia that Sharp ROM is based on as well as the X/Qt libraries which were not compiled with xscale optimisations. This means that you will be lucky just to get half the speed of pdaXrom since pdaXrom is quite optimised for each class of models, so if speed is what you want, then consider using pdaXrom instead. pdaXqtrom can be optimized a bit more but will never be as fast as pdaXrom since pdaXrom uses the faster soft float instead of the slower hard float which Sharp ROM is using.

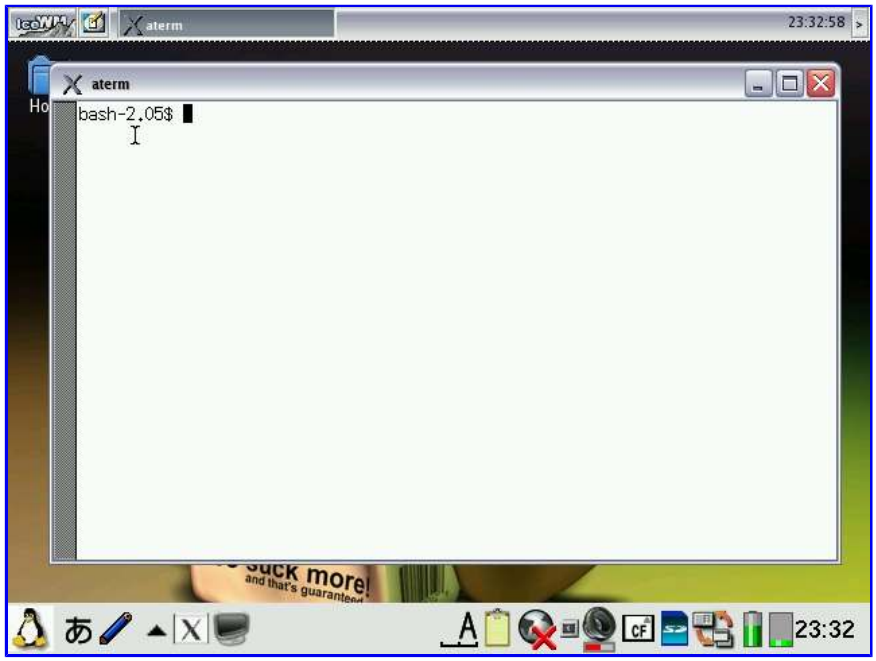
pdaXqtrom should also work on Cacko since it is a derivative of Sharp ROM. It should work on all clamshell models as well as Tosa. 5x00 models would struggle because of the limited memory and smaller screen size but should work too in a limited fashion (the previous X/Qt jumbo package would be a better choice for 5x00 models since it requires less resources). It is recommended to use a swap file of at least 64MB with pdaXqtrom. You can either manually create and enable a swap file/partition yourself, or use the qtopia-memoryapplet to do it. Cacko already includes a memoryapplet for creating swap files. I have packaged an improved version [qtopia-memoryapplet_1.0.4_arm.ipk] which allows C3x00 models to create the swapfile onto the internal MicroDrive on /hdd3 and also the creation of swapfiles up to 512MB.

The pdaXqtrom base image is available as a mountable cramfs and/or squashfs image, as well as a gzipped tarball which can be extracted to an ext2/ext3 filesystem. The default window manager for pdaXqtrom is **fluxbox**. Alternate window managers can be used also such as **icewm**. The file */etc/X11/defaultwm* specifies which window manager is loaded when X/Qt is started.

X/Qt running **fluxbox** with custom BluePDA theme



X/Qt running **icewm** with customised SilverXP theme



The X/Qt menu tab under Qtopia to launch X apps directly

A second image, pdaXqtrom apps, is also provided as an add-on with the applications listed below pre-installed and configured.

The pdaXqtrom base and application images are available from the following location: pdaXqtrom images

In addition, there also are a few special feeds for all the tested applications. The base and apps images are build from these feeds. The base image can be used with the pre-configured application image, or individual applications can be separately installed from the apps feed using the default Sharp installer.

The pdaXQtrom compressed images can be installed to anywhere you like. Basically, put them where ever you have the space for it. There are three different formats to choose from: cram, squash and tgz. The cramfs compressed images will work for everyone. In order to use the squashfs images, you will need to install a squashfs kernel module first. The advantage of squashfs over cramfs is that it has better compression and thus is smaller. The last option is a compressed tarball. If you don't want to use compressed loop back images but want to extract the files, then you can use the tgz files. However, make sure the destination for the install is an ext2 or ext3 formatted filesystem if you use the tgz file. For the cramfs/squashfs images, the filesystem type does not matter.

To install, copy the files you want to install and the install script (inside **pdaXqtrom-install.zip**) to the destination location. Make sure they are all in the same directory and then do the following:

```
# su
# ./pdaXqtrom-install.sh
```

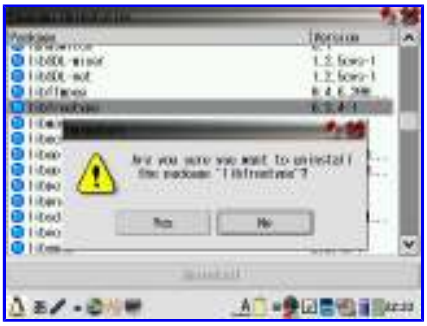
The install script will mount the cramfs or squashfs file images and run the setup scripts of each image. If you used the tgz file, it will extract the files and then run the setup scripts. Make sure you have sufficient space for the extracted files if you use the tgz files. The setup script does not check for sufficient disk space and just assumes you have enough free disk space.

You need to install at least the **pdaXqtrom-base** image. You can install the **pdaXqtrom-apps** image as well or install individual application packages manually. The **zgcc** image is an optional native C/C++ compiler if you want to compile your own applications on your Zaurus. It can build applications for QT/E 1.5 (Sharp/Cacko ROM) as well as X11 applications for pdaXqtrom (X/Qt). A minimal Perl is also included on this image. The **java** image contains **blackdown 1.3.1** as well as **jamvm 1.4.2** and **classpath 0.20** for running java applications. Several java compilers and tools such as **jikes** and **jar** are also included. **jEdit 4.2** has been configured to use blackdown jvm. **HdPad** (simple text editor) and **HdCrawler** (dynamic file downloader) are also included which use blackdown as well. The classpath samples can be tested by using the **jamtest** script. jamvm and classpath can be used to run applications compiled for java 1.4.x or later which blackdown cannot since it is only JRE 1.3.1. classpath is also used by the firefox java plugin (**gcjwebplugin**). You will need both, the **zgcc** and **java** images, if you want to use the firefox java plugin. Java is very memory hungry and very slow on the Zaurus and not all java applets will work properly either.

If you had installed X/Qt components before and they did not uninstall cleanly, then you can run

xqtcleanup which is located under the tools directory of the pdaXqtrom base image to remove them all before installing pdaXqtrom again. The xqtcleanup script only cleans up for the pdaXqtrom-base image.

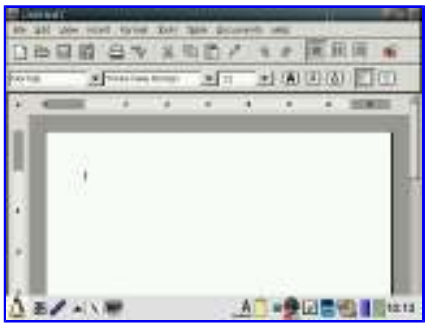
You should also uninstall **libfreetype** before installing pdaXqtrom if you have an older version already installed so that the pdaXqtrom installer can install a newer version of libfreetype. Simply uninstall the package with the Package Manager/Installer tool.



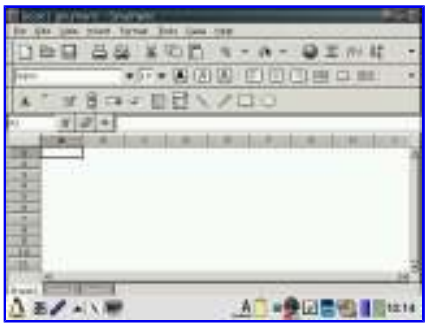
You might need to uninstall **libpng** as well. Alternatively, you can just upgrade libfreetype and libpng from the pdaXqtrom base feed after you have installed pdaXqtrom.

Note: You will need to remount the cramfs/squashfs images after each reboot. If you installed to SD/CF card, then you also need to unmount and/or mount them when ejecting/inserting the SD/CF card. However, if you install my latest automounter package, then this is automatically taken care of by the scripts. **automounter-c3000** has only been thoroughly tested for the C3000 and C3100 on the Sharp ROM. **automounter-lite** is a stripped down version of automounter and should work for all models. It only creates and mounts loop devices during bootup so it will automatically mount the images only if your card is inserted during bootup. automounter-lite is not as feature rich as automounter-c3000 but chances that it might break something or not work is much smaller.

The following applications are available on the pdaXqtrom-apps image and/or on the feed:

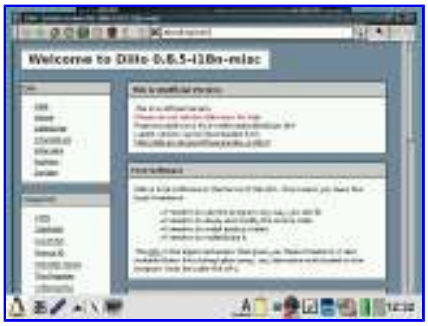


AbiWord 2.4.0



Gnumeric 1.6.0

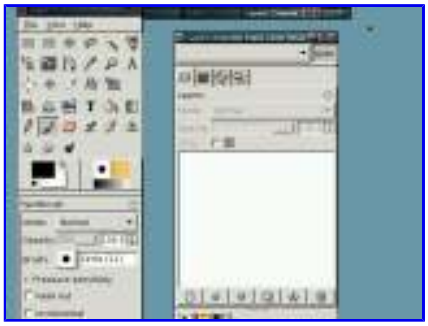
Firefox 1.5



dillo-xft 0.8.5

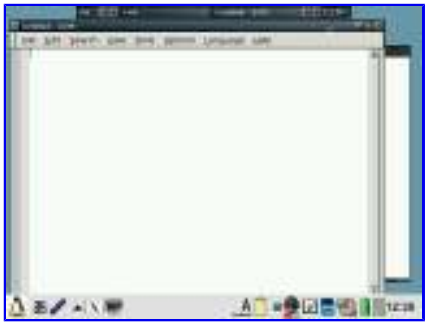
Thunderbird 1.0.7

Sylpheed 2.0.2

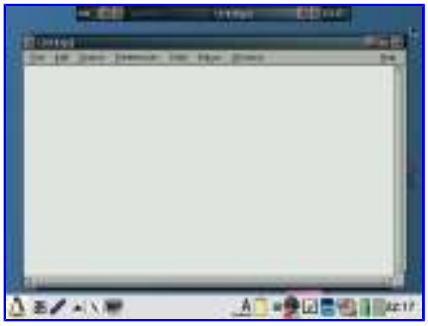


Gimp 2.3.4

GQView 2.1.1



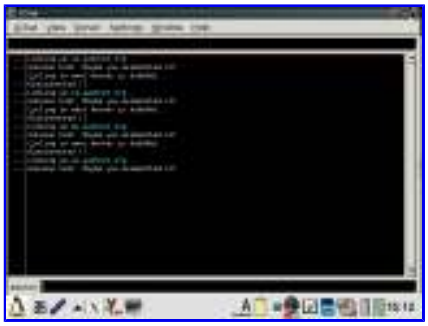
SciTE 1.62



nedit 5.4

gftp 2.0.18

axyftp 0.5.1



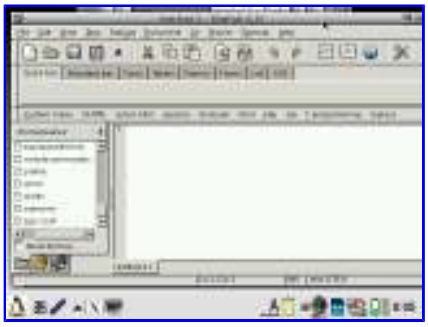
xchat 2.6

gaim 2.0

gyach 0.9.4

gtkyahoo 0.18.2

Dia 0.92



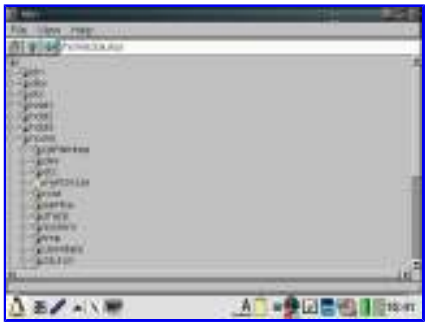
BlueFish 0.13

StarDict 2.4.3

Galculator 1.2.5

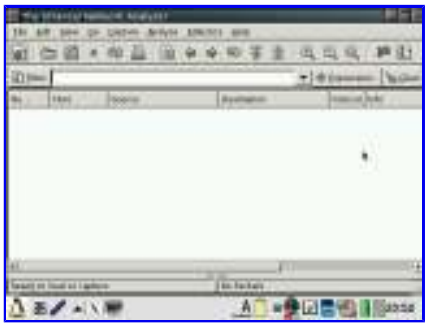
planner 0.13

fltdj 0.7



flfm 0.3

rox 2.2



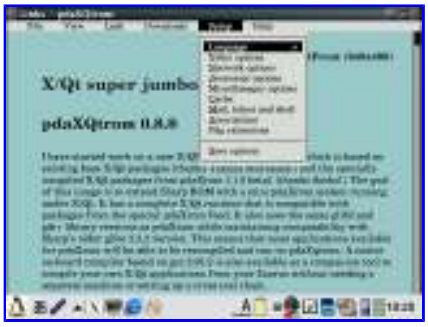
ethereal 0.10.9

tightvnc 1.2.9

epdfview 0.2.1.

xpdf 3.01

minimo 0.12



links 2.1

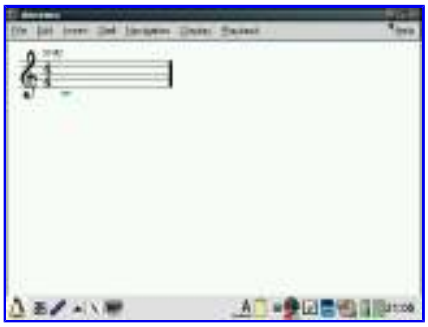
free42

x48

leafpad 0.7.9

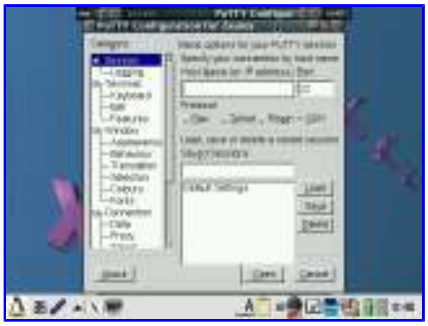
grisbi 0.5.8

ImageMagick 6.2.5

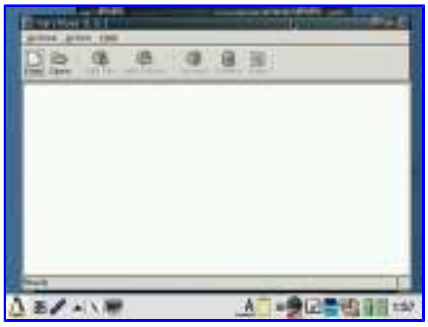


denemo 0.5.3

xmms 1.2.10



putty 0.58



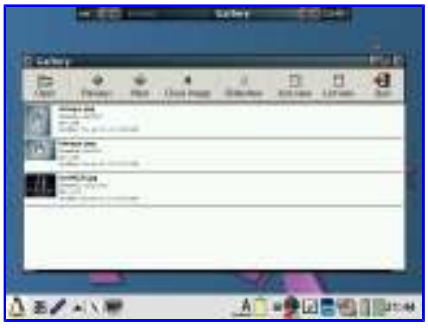
Xarchiver 0.3.1

mc 4.6.1

gpsdrive 2.10pre3

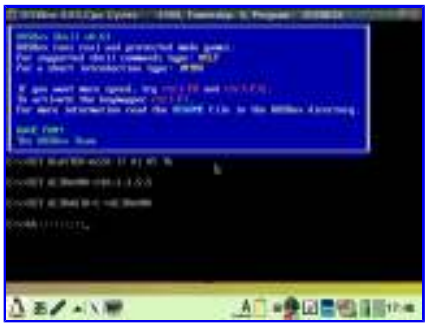
smsessy 0.1.1 (wip)

gpaint2 0.2.3

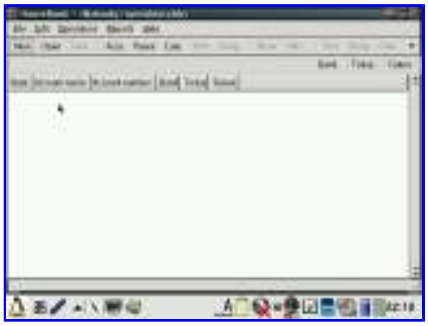


gpe-gallery 0.97

gmplayer 1.1pre8



dosbox 0.63



homebank 3.2

knowde 1.8.0

Many more untested packages are available from the pdaXrom X/Qt feed as well as the pdaXrom 1.0.5 feeds on the main pdaXrom site.

The pdaXqtrom images are installed as mountable filesystems (cramfs/squashfs). A setup script inside each image will create the required symbolic links and copy files appropriately. If you inadvertantly remove any of the links, you can simply rerun the setup script to recreate them. To uninstall, simply use the Package Manager/Installer tool and uninstall the pdaXqtrom package(s), then unmount the image(s).

You can also install and/or uninstall additional packages for pdaXqtrom using the Package Manager/Installer tool.

The main interface command for pdaXqtrom is **xlauncher**. It can be used to launch X application from Qtopia desktop scripts, the command line and even from within a X terminal. The general syntax is:

```
# xlauncher appname
```

You can also add a -debug right after xlauncher when you are trying to troubleshoot why an application is not being launched. The debug option makes xlauncher generate verbose output instead of hiding it:

```
# xlauncher -debug appname
```

There is also an experimental feature to embed the X application into the Qtopia desktop without having a full X window session. This feature is also known as rootless mode. Although the advantage is that it loads a little faster and consumes less memory, it also has drawbacks. The X application is not managed by any window manager and thus will not not be able to be resized or moved, nor are the corrective keyboard mappings available.

To launch an application in this mode, use the following command argument and syntax:

```
# xlauncher -embed appname
```

Here are some of the X applications in embedded mode:

PocketWorkstation can also be run under pdaXqtrom just like with the X/Qt jumbo package if the add-on deblauncher package is installed.

There are also some additional tools in the tools directory on the base image:

- mkqticon

  Use this script to generate a QT desktop icon for an X application if it does not have one. It will generate a launch script and desktop files under the XQt tab.

- mkfakepkg

  If you want to install applications directly from a pdaXrom feed, you need to update your package list and have all the packages that come with pdaXqtrom-base listed there. This script will generate the required info and populate the package list with it.

- monolingual

  This script can be used to enable or disable scim (Smart Common Input Method) which allows you to input other languages such as Chinese and Japanese. With scim disabled, ie monolingual enabled, X/Qt will startup faster. It is recommended if you only want to use English to enable monolingual mode. However, in order to input other languages in applications that support it, scim needs to be enabled. This can be done by disabling monolingual before starting X/Qt.

- xembed

  This script can be used to toggle the default behaviour of the Qt desktop icons. You can use it to enable or disable the xembed mode. When xembed is enabled, all the desktop icons under the X/Qt tab will launch the applications in embedded/rootless mode which means they run in a lightweight mode under X/Qt without utilising all the features of X and the application appears to be running inside Qtopia. If you disable xembed, then the desktop icons will automatically start the full X/Qt environment and run from within the full X/Qt environment.

- zhomefixC3000

  This script is for SL-C3000 users only. Since X applications place their config files under the user's home directory, ie /home/zaurus and the SL-C3000 only has 4MB on /home, it will get filled up pretty quickly. Run the zhomefixC3000 after you run each X application for the first time. The script will attempt to move the config files to /hdd2/zaurushome and symlink it back.

- xqtcleanup

  This script usually is not required and should not be run. It is a brute force approach to remove existing X/Qt files. Only use this script if you had a broken X/Qt installation previously that you could not cleanly uninstall or you get other errors.

- cackofix

  This script may be needed to be run by Cacko users if they get glib errors. If everything works, then don't run this script.

X/Qt has its own set of basic key controls. The *menu* key is hardcoded to activate the X/Qt control which allows you to activate fullscreen mode or shutdown X/Qt. *Fn*+*m* shuts down X/Qt immediately. Also *Fn*+tap is equivalent to right mouse click whereas a single tap with the stylus is a left mouse click. Sometimes, when switching between X/Qt and QTopia, the caps lock is turned on under X/Qt. Just hit *Shift* several times to disable it.

The rest of the keys is controlled by the keymapping file located under /etc/X11/kb/xmodmaprc. This file is loaded each time X/Qt starts up. During installation, an appropriate mapping file has been copied there depending on your Zaurus model. If you want to customise it, then copy it to /home/zaurus/.xmodmaprc and edit that copy in your home directory. You can use the **xev** command to determine the keycode for each key. If you have a Cxx00 model, ie one with 2.4.20 kernel, and you are using my custom keyhelper.xml, then you might also want to use my custom xmodmaprc file as well instead of the standard xmodmaprc. To use my custom xmodmaprc copy /mnt/pdaxqtrom-base/config/settings/xmodmap/xmodmaprc-custom to /home/zaurus/.xmodmaprc

In addition, FluxBox also has some extra key shortcuts defined in /home/zaurus/.fluxbox/keys. Most of those shortcuts are defined to use the Mod4 (assigned to Super_L) and Mod1 (assigned to Alt_L) modifier keys which I mapped to the Home and Alt (left kanji) key respectively. FluxBox menu is activated by holding the Fn key and tapping on the desktop. To release/cancel from the menu, tap anywhere on the desktop (but not on any window/application on the desktop). You might also need to press the *Cancel* key as well.

Here are the key shortcuts for FluxBox:

- Alt + Tab = Next Window
- Alt + Shift + Tab = Previous Window
- Alt + c = Close Window
- Alt + m = Minimise Window
- Alt + n = Maximise Window
- Alt + h = Maximise Window Horizontally
- Alt + v = Maximise Window Vertically
- Alt + s = Shade Window
- Alt + t = Toggle Decor
- Alt + r = Raise Window
- Alt + l = Lower Window
- Alt + Up = Move Window Upwards
- Alt + Down = Move Window Downwards
- Alt + Left = Move Window to the Left
- Alt + Right = Move Window to the right
- Alt + space = FluxBox Menu
- Alt + 1 = Virtual Window 1
- Alt + 2 = Virtual Window 2
- Alt + 3 = Virtual Window 3
- Alt + 4 = Virtual Window 4
- Alt + x = Launch Terminal

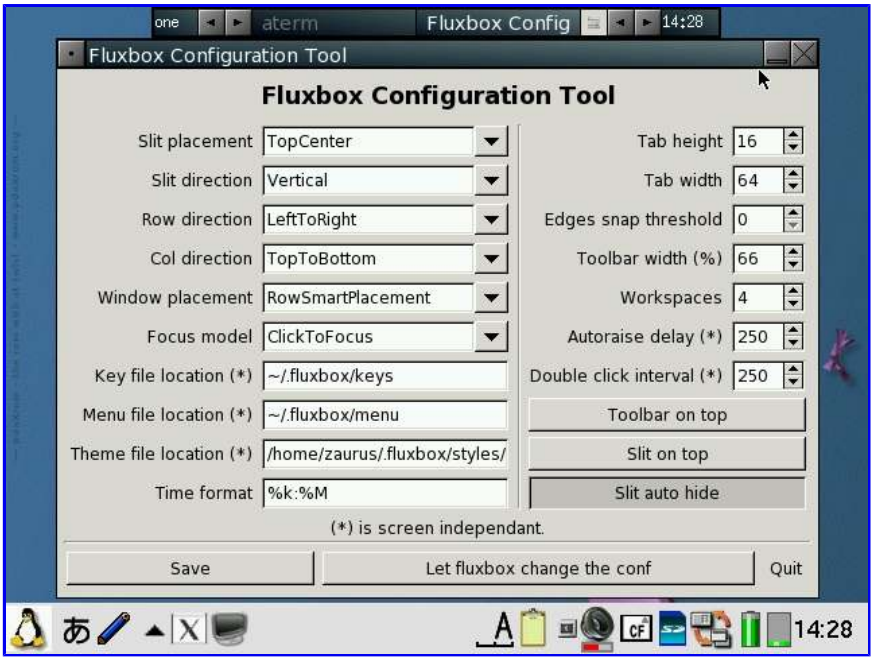
multi-aterm has the following key shortcuts:

- Ctrl + Alt + n = New Tab
- Shift + Left = Previous Tab
- Shift + Right = Next Tab

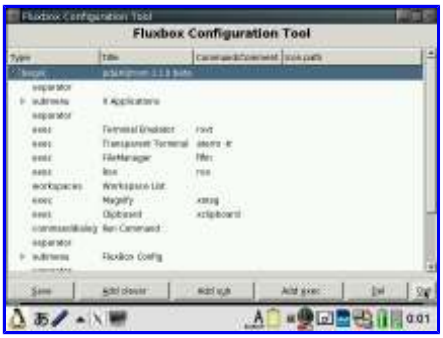
You can also customise the theme and styles. FluxBox comes with a set of styles located under /opt/QtPalmtop/share/fluxbox/styles. I have created one called BluePDA which is used by default. You switch styles easily from the FluxBox menu under the FluxBox Config options. To customise a style, copy it to /home/zaurus/.fluxbox/styles and edit away. The background image is part of the style, so in order to change the wallpaper, you will need to edit the style unless you are using the default BluePDA style. Its wallpaper is a symbolic link to /opt/QtPalmtop/share/backgrounds/wallpaper.jpg and you can just replace it with any jpg file.

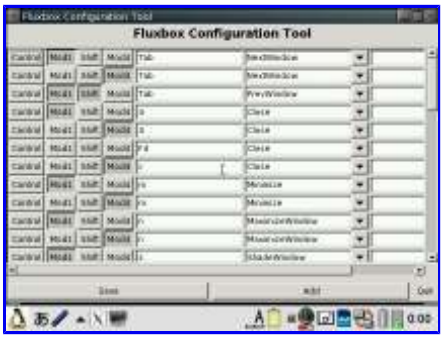
There is also a GTK+ theme selector to switch GTK+/GTK2 themes and styles. I have added a few themes and they are located under /opt/QtPalmtop/share/themes. You can copy additional themes to that location.

You can customise FluxBox, its menu and shortcut keys using the FluxBox config tools.

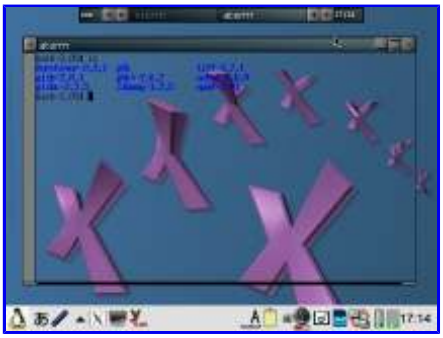


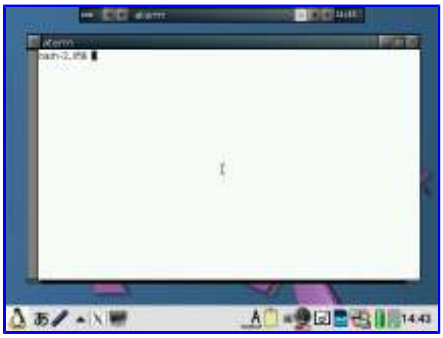
The menu items can be customised with the FluxBox menu editor. The shortcut keys can be customised using the FluxBox key utility.



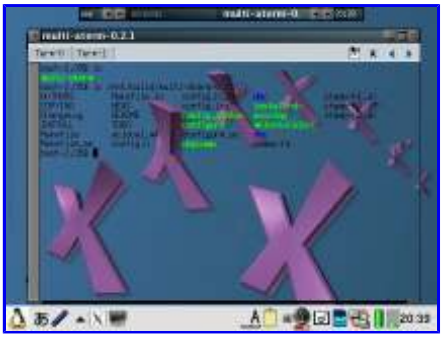


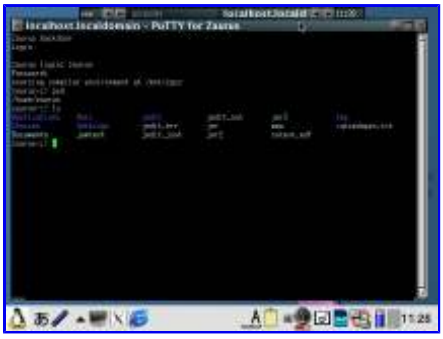
A transparent terminal (aterm) is also available which you can also use in opaque mode if you prefer.





Additionally, a tabbed multi-aterm is also available and can be used in both opaque and transparent mode. You can also use the putty terminal as well.

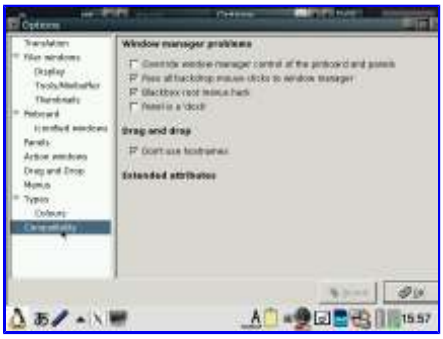




Desktop icons are provided by idesk. You can manually create additional desktop icons or use the

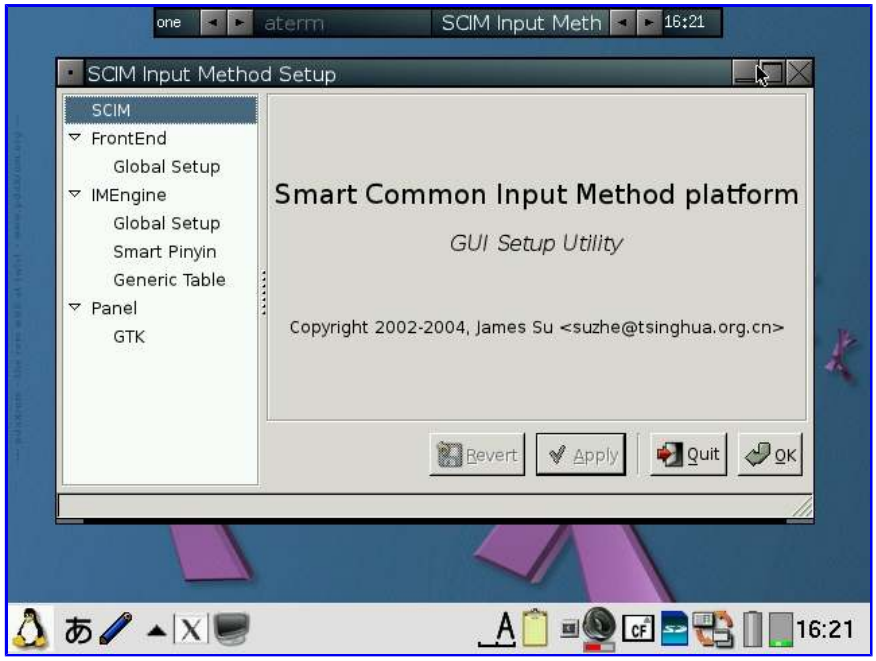
genicons script which is also available from the FluxBox config menu to generate the icons. This script will try to duplicate all the Qtopia desktop icons under the X/Qt tab. You can enable and disable the display of these icons from the FluxBox config menu as well. The idesk icon definitions are located under /home/zaurus/.idesktop

Alternatively, you can also use rox pinboard to create desktop icons. This is also available from the FluxBox config menu. However, the rox pinboard will take over the control of the menus and you will not be able to access the FluxBox menu anymore unless you configure rox to run in blackbox/fluxbox compatability mode.

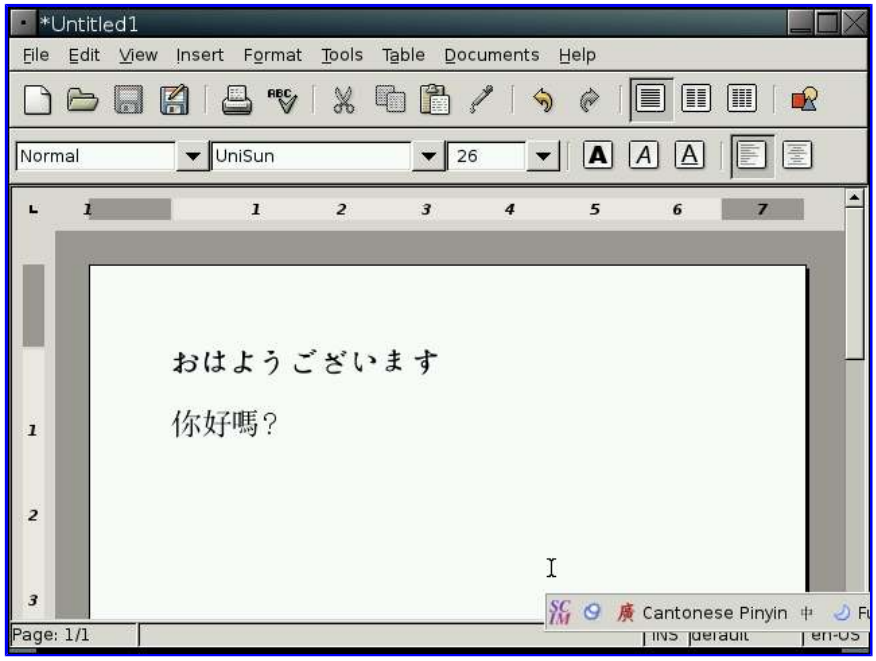


You will need to check both "*Pass all backdrop mouse clicks to window manager*" and "*Blackbox root menus hack*" in order to get the FluxBox menu back.

pdaXqtrom also has multilingual support built-in as well. It has uim and anthy, as well as scim embedded. Some Japanese and Chinese fonts are also included. You can add additional fonts by copying them to /usr/share/fonts or .fonts under the user's home directory, ie /home/zaurus/.fonts



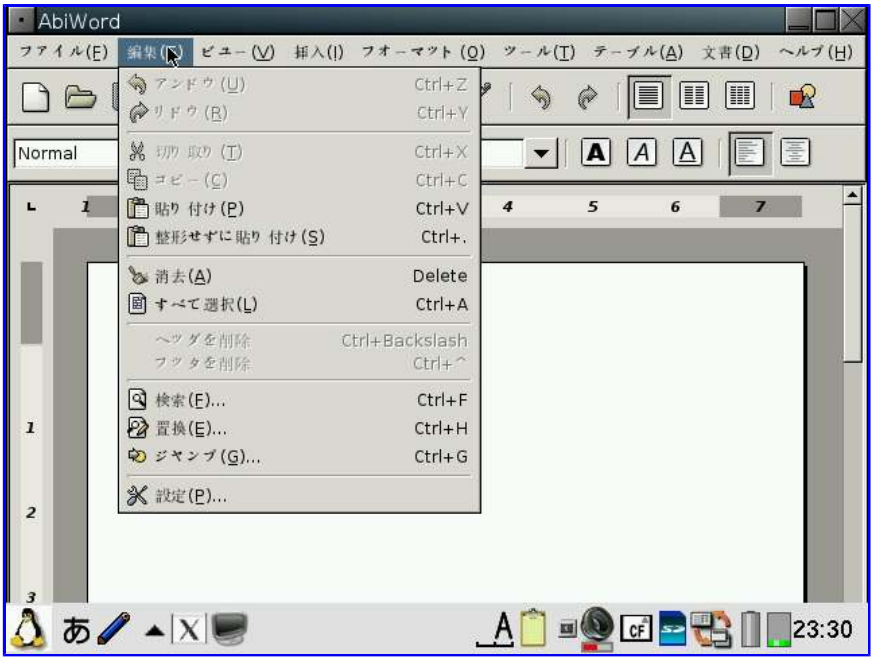
scim is activated by pressing *Ctrl+space*. You will get a scim menu at the bottom of the screen. You might need to switch X/Qt to fullscreen mode to see it. From the scim menu, you can select the input language you want to use. I have customised it for Chinese and Japanese primarily, but other languages can also be used with it.



If you want to prevent scim from loading and thus start up faster and use less resources, then use the *monolingual* command or create a file called "monolingual" under /etc/X11:

```
# su
# touch /etc/X11/monolingual
```

The applications for pdaXqtrom are mostly the same version as the ones from pdaXrom. However, I tweaked and hacked a few of them to make them more user-friendly.



Abiword has been configured to start in 640x480 or 480x640 mode depending on the model and display English even on Zauri with Japanese locales. Abiword uses the system locale to determine what language to display menus in, ie English menu with en locale, Japanese with ja locale. However, since many clamshell models on Sharp ROM use the ja locale with the menus converted to English, the same was required for AbiWord. The file /etc/X11/app-defaults/Abiword.strings contains the text for the menus and is a copy of en-GB.strings for English menus. If you want your menus in another language, then simply replace it with the language file of your choosing. I also compiled aspell and included a dictionary so you also get an English spell checker. Additionally, I also compiled a few plugins for abiword, mainly text converters so you can open additional document types such as OpenOffice, MSWrite and PDF files. I also got printing to a PDF file working partially. Currently it only works with certain fonts, so if the results looks like garbage, try again with another font.

Firefox has been enhanced a bit. It originally had a small problem with its own launch script but now it launches itself into a new tab if you launch firefox multiple times when it has already been started. I have also compiled the gplflash plugin so flash animations can be viewed with firefox. A java plugin is also available, but it requires jamvm and classpath (available on the java image) and perl (available on the zgcc image).

GQView, xpdf, grisbi, putty and abiword have been recompiled to force them to stick to the 640x480 screen size and not go beyond. Also, most applications have been customised to start in full screen but not covering the bottom menu bar. This has been done for both portrait as well as landscape based models and is managed by FluxBox. I also hacked GTK2 to restrict the window size to 640x480 and 480x640, so hopefully all applications will fit onto the screen. I have tested AbiWord and FireFox and all dialog boxes I could find were correctly sized. However, if there are any oversized windows, then use *Alt*+*n* to maximise the window or *Alt*+arrow keys to move them around. I purposely hacked GTK2 instead of the window manager (fluxbox) which is where it should be fixed. However, since X/Qt can run in rootless mode without a window manager, the window size needs to be controlled higher up in GTK2.

A rotate feature which resizes the windows of running applications has also been added and is accessible from the FluxBox menu.

StarDict has not been configured with any default dictionaries. You will need to install your own dictionaries into /usr/local/share/stardict/dic. Alternatively, I have created a huge squashfs image with many stardict and qbedic dictionaries as well as the stardict sound files for pronouncing the words in English. You can use that image (dictionaries.squashfs) as well but you will need more than 300MB of space for it.

xmms has been hacked to allow it to use more Zauri friendly keys. You can now use the *OK* and *Cancel* keys as Play and Stop. Also *Up* and *Down* have been reassigned to change tracks and *N* and *M* are now used for volume control. The goal was to be able to use the sidekeys on the back of the Zaurus to control xmms just like the MusicPlayer when the display is closed or you are using it in PDA mode (portrait) without access to the keyboard. xmms can play wav, mp3 and mod files and I have also compiled mplayer 1.1pre8 with X11 support so xmms can launch mplayer inside a little gtk window for watching short clips.

I have also compiled the mplayer GUI - gmplayer. Note that this version of mplayer runs via X/Qt which runs via a frame buffer through QT/E so it will never be as fast as mplayer running via bvdd or ati. However, mplayer is compiled with XScale CPU optimisation so it does perform reasonably for small clips and it supports more codecs than the older version.

A hacked version of DOSBox is also included. The **:** key has been hacked to work in this version. **Fn** + **;** will give the **:** key. **Fn** + **Ctrl** + **;** will give the **;** key. To exit dosbox just type **exit**.

In addition, the following tools are available via the command line in pdaXqtrom:

pdf tools

- pdffonts
- pdfimages
- pdfinfo
- pdftohtml
- pdftoppm
- pdftops
- pdftotext

video tools

- swfplayer
- mplayer2

sound tools

- esdloop
- esdplay
- esdrec

internet tools

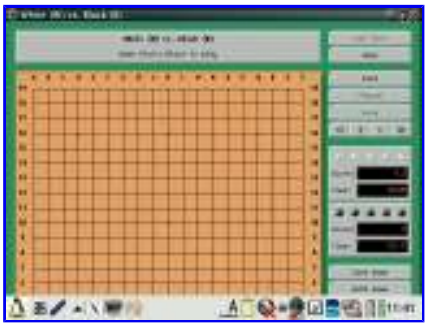
- links
- gftp-text
- plink
- psftp
- pterm
- puttygen
- puttytel

graphics tool

- animate
- composite
- convert
- identify
- import
- mogrify
- montage

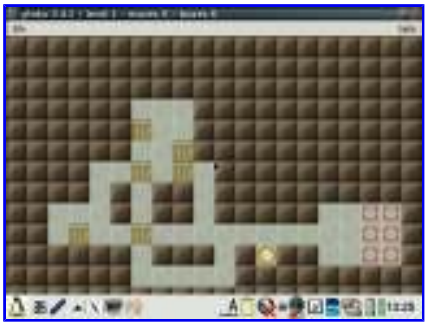
pdaXQtrom Games packages

The following games are available on the **games** image and/or on the extra games feed:



GNU go 1.9.12 (xlauncher cgoban)

GNU chess 5.07 (xlauncher chess)



gsoko 0.4.2 (xlauncher gsoko)

gtkatlantic 0.4.0 (xlauncher gtkatlantic)

xbomb 2.1a (xlauncher xbomb)

xdemineur 2.1.1 (xlauncher xdemineur)

xkobo 1.11 (xlauncher xkobo)

xpuyopuyo 0.9.8 (xlauncher xpuyopuyo)

xshogi 1.3 (xlauncher xshogi)

xsokoban 3.3 (xlauncher xsokoban)

Alternate Distros/ROMs:

Beside the Qtopia 1.5 based Sharp distro which is often wrongly referred to as Sharp ROM, there are several other distros available for the various Zaurus models. However, at the moment, the Sharp distro is the only really stable distro available for the SL-C3000. There is currently work underway to convert the various ROMs into a distro that runs on the SL-C3000, but realistically, it will be a while until stable versions of those will be available. OpenZaurus 3.5.4.1 is the only other Linux distribution that officially supports the SL-C3000.

Support for the SL-C3100 version of those other ROMs already exist even though the SL-C3100 is newer than the SL-C3000. This is because of the larger flash the SL-C3100 has which allows it to be flashed with a ROM image similar to all the other Zauri models that run off the internal flash. In fact, the SL-C3100 is treated like a SL-C1000 with an extra internal MicroDrive and hence support for the extra MicroDrive is miniscure.

The SL-C3000 is the only model that was designed to run off the internal MicroDrive (harddisk) instead of flash. Therefore, extreme care should be taken when putting an alternate distro onto the SL-C3000 since it cannot be easily backed up with a NAND backup like the other models. Only do it if you know what you are doing and make sure you know how to recover if something goes wrong. Restoring a SL-C3000 is not as easy as restoring the other models either. Stick to the Sharp distro if you are a newbie. Advanced users still should be very careful when they try out the other beta distros. Keep in mind that putting another distro onto the SL-C3000 will wipe everything on it so make sure you make appropriate backups before doing so.

Since I have two Zauri, both the SL-C3000 and the SL-C3100, I can make one of them my stable working mini laptop with all my customised settings, applications and data while the other one will be used for application development for the Z and trying out other ROMs/distros. Ultimately, I will need a better alternate distro for my SL-C3000 since the default Sharp ROM or Cacko are too limiting for its small 4MB home partition. My SL-C3100 on the other hand is functioning perfectly with a customised Sharp ROM and X/Qt (see pdaXqtrom for details on running a host of X applications on the default Sharp distro).

Cacko

The Cacko distro is an improved Sharp distro which basically fixes a lot of the shortcomings of the Sharp distro. Also, Sharp distro is by default completely in Japanese. The Cacko distro is essentially Sharp distro completely translated into English with updated driver support.

The Cacko 1.23 release was mainly aimed at the SL-C1000, with the SL-C3100 being treated just like a SL-C1000 with an internal CF. Nothing is put on the internal MicroDrive, so it remains untouched by the Cacko installation. Cacko 1.23 is released in two editions, the full edition and a lite edition.

The lite edition is a barebone distro where most the pre-installed applications and modules have been removed to give you the maximum of available space on the flash. All the left out applications and modules can be installed from the Cacko feed. They all have a *lite* in their package names for easy identification.

The full Cacko 1.23 distro is very similar to the Sharp distro. It enhances the kernel with patches similar to the Tetsu special kernel and also adds additional filesystem support such as squashfs, hfs, iso/joliet, ntfs and fuse. The main difference between the Cacko ROM and the Sharp one is that it is completely in English and all the Japanese components, including the dictionaries have been removed. Also, a lot of work has been put into networking and bluetooth support, thus a lot of the network and bluetooth adaptors, both CF based ones as well as USB dongles, are automatically recognised and just work with Cacko without the need of installing and configuring drivers for them. Some of the default applications that come with the Sharp distro have also been replaced with better alternatives, in particular the ones that were Japanese only applications. In addition, a number of essential Linux tools and Qtopia based applications have also been added and are available on Cacko pre-installed. See the Cacko release notes for a detailed list of added features.

To install Cacko, you would have to flash your SL-C3100 with the Cacko ROM. This will wipe your entire flash, so make sure you do a Nand backup so you can restore your Zaurus to the previous state in case you want to revert back, but also make a backup of your data if it is stored on the flash since it gets wiped. The /hdd3 partition on your internal MicroDrive won't be wiped by flashing the Cacko ROM so it is not essential to back it up prior to flashing Cacko, but it would be a good idea anyway just in case

something goes wrong. You do not need to resize the root partition for the SL-C3100 because 32MB is the default root partition on the SL-C3100 unles you have changed it to something else previously. The SL-C1000 has a default root partition of 64MB so repartitioning it on the SL-C1000 will free up extra space for your user partition to install applications and to store user data.

Thus for the SL-C3100, just flash the ROM without repartitioning root, ie use the *Install new ROM* option straight away and say **yes** to *format user partition*. If you say no, just your root partition will be reflashed and you get the Cacko kernel and config, but all your application binaries will remain unchanged. Select **Reboot** when the flashing has finished and you are done.

There is a Cacko 1.23 beta available for the SL-C3000. Since the SL-C3000 has a very small flash partition, installing another distro will wipe your entire MicroDrive. You won't be able to simply make a Nand backup and restore your system if you want to revert back. You would need to rebuild your SL-C3000 from scratch. However, the C3000 beta release of Cacko leaves your partition table layout as is and just reformats your partitions. Be aware that the C3000 beta is quite different from the normal 1.23 release of Cacko, it does not upgrade the kernel nor does it provide all the extra modules (since they won't fit onto the tiny flash where modules normally are located).

I only tried the full Cacko 1.23 on my SL-C3100 and noticed quite a few annoying error messages when booting up Cacko on it (more than the usual ones you get for Sharp with the Tetsu special kernel installed) but it does boot up and it works.

I have hacked the startup script (rc.rofilesys) to eliminate some of the errors since they are a thorn in my eyes, and I have also applied the same hack that I applied to the default rc.rofilesys for the Sharp distro. The hacked script allows you to boot the SL-C3100 without needing /hdd1 and /hdd2, but it will attempt to mount /hdd1 as swap and /hdd2 as ext3 if they exist.

Customising Cacko is similar to customising the Sharp distro, but a bit less customisation is required since a lot is already pre-customised. For example, LUSScreensaver, mplayer and kino2, ko/pi, opera, mc, ssh and sudo are already pre-installed if you install the full Cacko edition, just to name a few.

However, there are also a few differences introduced by Cacko. The startup image in Cacko is bz2 compressed to save space. It gets extracted to /tmp (which is 10MB in size compared to the 1MB on Sharp) during startup and loaded from there. Once loaded, the extracted image is deleted. A bigger tmpfs allows you to open larger files but since tmpfs uses physical RAM, less physical memory is available, so instead of having 63MB, you end up with 54MB. This is a reasonable trade-off.

Cacko also includes an updated battery and memory applet which allows you to generate and manage swapfiles of sizes up to 128MB (the previous version was limited to 64MB only, however, there is a newer Japanese version which allows swapfile sizes up to 512MB) and overclock as well as underclock the CPU. Furthermore, Cacko does not really use rc.rofilesys to mount the internal MicroDrive partitions. Instead, it remounts everything using the /sbin/hddmount script which is called by /home/QtPalmtop/qpe.sh after all the rc scripts have run. You might as well put the script to load the SD card service after hddmount and eliminate the S04sd links so you won't get flooded with MMC warnings during bootup if you have your SD card inserted.

First disable SD driver from being loaded by the rc scripts:

```
# mv /etc/rc.d/rc3.d/S04sd /etc/rc.d/rc3.d/_S04sd
# mv /etc/rc.d/rc4.d/S04sd /etc/rc.d/rc4.d/_S04sd
# mv /etc/rc.d/rc5.d/S04sd /etc/rc.d/rc5.d/_S04sd
```

Then modify /home/QtPalmtop/qpe.sh to load the SD driver instead:

```
/sbin/hddmount
while true ; do

    if [ -f /home/QtPalmtop/pics144/Startup_screen.bmp ]; then
        sdisp /home/QtPalmtop/pics144/Startup_screen.bmp &

    else
        bzcat /usr/QtPalmtop.rom/pics144/Startup_screen.bmp.bz2 >
        /tmp/Startup_screen.bmp
        rm -f /tmp/Startup_screen.bmp
```

```
fi
/etc/rc.d/init.d/sd stop
/etc/rc.d/init.d/sd start
cd
```

Since the latest Cacko (1.23) is not available for the SL-C3000 except for the beta version which does not update the kernel and add additional drivers anyway, I have extracted the features that I really like in Cacko and created packages that can be installed on the SL-C3000 Sharp distro so that my SL-C3000 can also benefit from some of the useful Cacko features and updates that is available for the SL-C3100 without having to install the beta which is just missing too many features provided by the proper Cacko 1.23 to make it worthwhile.

Here is a list of packages extracted from Cacko with short descriptions:

- **qtopia-sysinfo_1.23_arm.ipk** - enhanced sysinfo tool with process and mount controls
- **qtopia-addressbook_1.23_arm.ipk** - addressbook with alphanumeric sorting support
- **qtopia-combbatteryapplet_1.0.6_arm.ipk** - updated battery applet with overclocking/underclocking support
- **qtopia-memoryapplet_1.0.3_arm.ipk** - updated memory applet with better swapfile management
- **qtopia-keyboardapplet_1.0.0_arm.ipk** - keyboard layout mapper applet
- **qtopia-network-usblan_1.0.0_arm.ipk** - network config for usb lan adaptor
- **qtopia-network-bluetooth_1.0.0_arm.ipk** - network config for bluetooth adaptor

The following are some of the driver packages available from the Cacko feed to add the same driver and networking support as Cacko:

- vga-console-font_1.0-1_arm.ipk - vga console font
- input-modules-2.4.20_1.23-lite-1_arm.ipk
- isofs-modules-2.4.20_1.23-lite-1_arm.ipk
- hfs-modules-2.4.20_1.23-lite-1_arm.ipk
- nls-modules-2.4.20_1.23-lite-1_arm.ipk
- ntfs-modules-2.4.20_1.23-lite-1_arm.ipk
- udf-modules-2.4.20_1.23-lite-1_arm.ipk
- usb-camera-modules-2.4.20_1.23-lite-1_arm.ipk
- usb-network-modules-2.4.20_1.23-lite-1_arm.ipk
- usb-storage-modules-2.4.20_1.23-lite-1_arm.ipk
- iptables-base-1.2.11-lite-1_arm.ipk
- iptables-extras-1.2.11-lite-1_arm.ipk
- iptables-modules-2.4.20_1.23-lite-2_arm.ipk

If you don't want to flash Cacko, but want the additional and updated functionality that Cacko provides, then you can install the Tetsu kernel and modules followed by additional Cacko lite modules (2.4.20). This should give you a kernel similar to Cacko. Then you can install the Cacko lite applications from the Cacko feed and the extracted packages mentioned above. This would get you very close to the functionality and features of Cacko 1.23.

When customising Cacko with the customisation guide for Sharp distro, some packages and instructions can be skipped, eg. in the essential section Tetsu kernel, keyhelper, sudo and qkonsole are mentioned, but Cacko already contains them so it is not required to install them again. However, the keyhelper and automounter config in Cacko is very minimal and installing the custom config for those will enhance the capabilities for them.

You can also add Japanese support back into Cacko by installing the [japanese-support-c3100jaen_1.23_arm.ipk] package and copying the dictionary back from the CD-ROM as well as installing the zdict and translater packages. The japanese-support-c3100jaen package which I have created does not restore all the Japanese menus. It gives you the Japanese inputmethod support without changing all the messages and the addressbook back into Japanese so the result is similar to what my custom-jaen package does on the Japanese Sharp distro.

The above is a screenshot of Cacko after heavy customisation. It looks just like my customised Sharp distro. I also symlinked an Australian locale/inputmethod and also added an Australian icon for the keyboard applet.

pdaXrom

pdaXrom is another alternate distro for the Zaurus. It is essentially a complete X windows environment. Thus pdaXrom would make your Zaurus into a mini laptop with an X server and a lot of the X applications available on your desktop/laptop Linux distro ported to the Zaurus. There is also a pdaXrom X86 version that is for your desktop/laptop so you can run the same applications on your Zaurus and your PC.

The latest pdaXrom release is currently 1.1.0 r121 and the SL-C3100 is supported, however, it is not stable and full of bugs. There is also a beta version for the SL-C3000 based on the 1.1.0beta1 release (pdaXrom 1.1.0 beta1 for C3000 beta2) which is much more stable. There are still a few issues with the betas and hopefully, once they are all resolved, pdaXrom release version should be great. The most stable versions are still beta1 and beta3. Anything after that, ie beta4 or r121 are stil in the early stages of development and testing.

I decided to test pdaXrom on my SL-C3000 because the C3000 beta2 release of pdaXrom 1.1.0 beta1 features a pivot boot feature that uses /home on the MicroDrive instead of the tiny 4MB /home on the flash. This neatly fixes the problem of the tiny /home running out of space since the flash on the C3000 is only 16MB and only 4MB of it is allocated to /home. However, this also prevents the MicroDrive to spin down for power saving and thus battery time is reduced, but it also responds faster due to no repeated spin up times (just replace the MicroDrive with a Flash Card and it will be perfect).

Installation went flawlessly following the instructions on the OESF forum. The C3000 installer repartitions and reformats the entire MicroDrive into a single 4GB partition during install. Once installation is finished and you reboot, you will end up at the command line login prompt. Login as root without password and then use **startx** to get into the X GUI (well, it should boot right into the GUI since it is runlevel5).

pdaXrom is a X windows environment that is by default pre-configured to use openbox as its window manager and uses matchbox applets for its panels and taskbar features. There is also a host of default applications pre-installed with the base pdaXrom install, most of which are quite useful.

However, there are many more applications available for pdaXrom from its feeds. The X86 version which can be installed on a PC or Laptop has all the available applications pre-installed by default. The C3000 and C3100, as well as the C1000 and C3200 share a common feed. In addition, there is also a generic feed marked as C7x0 feed which is also required. To install more applications, use the Package Manager and configure the feed locations. The default feed URL pre-configured in the Package Manager is out of date and should be replaced with the generic (C7x0) and C3x00 specific feeds. You can also configure local feeds. Just download the feed.tar.gz files and extract the content to a card or the internal disk. Then point the package manager at the directory containing the feed contents (eg: file:///home/root/Documents/installs/C3000/feeds).

The following applications were already pre-installed:

- SciTE
- AbiWord
- xpdf
- dillo
- licq
- sylpheed
- xchat
- xmms
- GQview
- xdemineu
- aterm
- galculator
- xclock
- xircp
- xterm

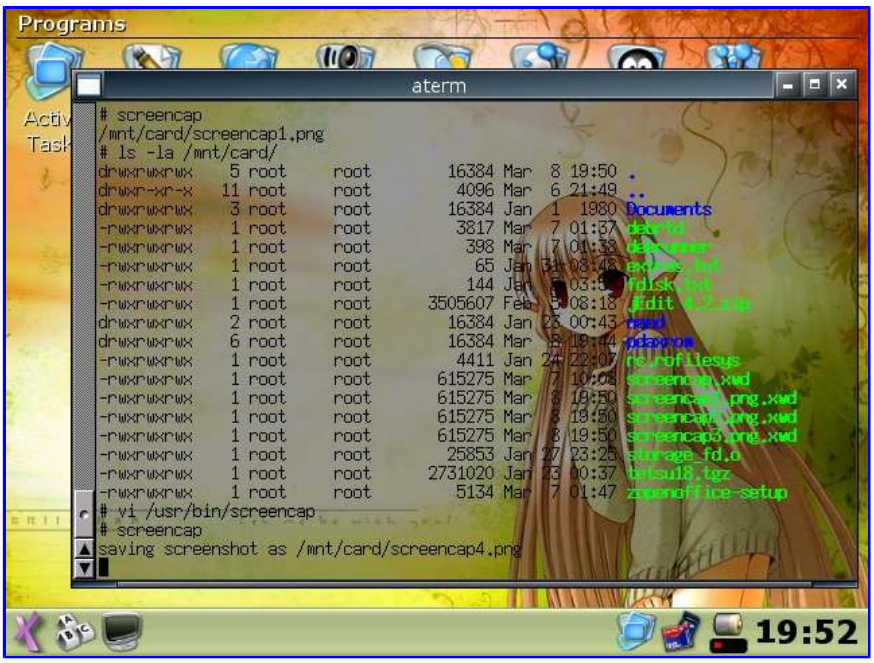
This is quite a good start, but to be really useful, there are a few extra applications and utilities needed. I have added quite a few packages to make it as functional as a full laptop. See the pdaXrom customisation section for more details.

So far pdaXrom seems quite fast. It looks very nice also and seems to be more responsive than Qtopia. I am very impressed with the speed. I can run both quake and quake2 in fullscreen (320x240 mode). This makes me very happy since Doom2 and Quake2 are my favourite games and having them run on the Zaurus is just great!

There is also an on-board compiler available for pdaXrom which can be used to compile your own packages or develop new applications.

There were also a few minor problems I encountered with the C3000 beta2. The boot process is much cleaner than the Sharp distro and it looks much more like a Linux boot process. Yet, there were a few small annoying error messages which would had otherwise made it perfect. Those error messages were easily fixed. The default keyboard mapping is a bit odd for my taste so I customised it to my liking. Also, the suspend feature seems to be a bit buggy.

Some of the applications from the feeds need to be customised before they can be used effectively. AbiWord for example, installs with a default screen size that is larger than what the C3000 and C3100 have (640x480), so you won't see part of the application that extend beyond the screen. This can be fixed by manually editing the default config files for those apps and changing their default screen size.



This is a screenshot from the pdaXrom 1.1.0 beta1 on the C3000 after a bit of tweaking with the look and feel.

See my custom pdaXrom page to see a list of customisation/fixes that I have applied to my C3000.

The bad thing about pdaXrom is that after beta3, things go downhill. For each bug fixed a new one is introduced. The beta4 and its successor r121 are really new versions because of their drastic change with the introduction of a new boot mechanism and replacement of the stable 2.4.20 kernel with the faster but buggier 2.6.x kernel. As a result of this, most configuration tools are broken and many of the applications need to be recompiled. If you want to put up with a buggy distro for testing, then you might as well move to OpenZaurus instead because it actually fully supports all C3x00 models.

OpenZaurus

OpenZaurus, or OZ as its commonly known as, is yet another popular Linux distro for the Zaurus. OpenZaurus comes in two main flavours which you can choose from. There is OPIE and GPE. The OPIE flavour of OZ is an opensource implementation of a newer version of Qtopia and a likely candidate as a replacement for Sharp ROM/Cacko. OZ/GPE is the X11 flavour of OpenZaurus and similar to pdaXrom. There is also an experimental Enlightment version too which is another powerful X11 window manager. Finally, there is also OZ/bootstrap which is just the bare minimum bootable image of OZ.

You can also install Opie and GPE on different virtual terminals and thus have both running at the same time and switch between the terminals. This can be done with a bit of hacking but should ideally be the default for the clamshell models so you get the best mix of applications from both Qtopia and X11 environments since they have plenty of space for it.

However, OZ appears to be very developer centric. It is coupled with a build system, OpenEmbedded, and very much focused on building, packaging and source control. It is a system build from the ground up and focuses a lot on the kernel and clean builds. The OZ team appears to be very organised and have a very structural approach. Unfortunately, this sometimes comes across as being inflexible and stubborn by some end users. It also sometimes appear that usability is given less importance in favour of portability. The gap for this is filled with the Hentges distro, which is a more user-friendly distro based on OZ. Hentges is pretty much to OZ what Cacko is to the stock Sharp ROM. Both are a more user-friendly repackaging of a more rudimentary distro. Hentges is currently available for Akita (C1000) and Spitz (C3x00).

The latest version of OpenZaurus for the C3x00 is 3.5.4.1 which is build with the 2.6.16 kernel instead of the standard 2.4.20 kernel that the Zaurus was released with. OpenZaurus being a bleeding edge distro, with lots of active development, has implemented many more advanced features than the Sharp distro, however, as with all bleeding edge technologies, backward compatability and stability is not always guaranteed and thus certain new features might break existing features and/or applications inadvertantly. This is something very hard to avoid for software projects unless you take Sharp's approach of not updating the environment and just patching it up when forced to. OZ takes the opposite approach. It is constantly being build and it integrates a lot of features from other Linux distros such as altboot and kexec. At the moment, TCP/IP via USB client cable does not work when connecting to Windows. This is the biggest problem I have with OZ since I like to plug it into my Windows PCs.

OZ also takes advantage of the MicroDrive available on the C3x00 and thus installs itself to the MicroDrive instead of the flash. This of course has its advantages and disadvantages. On the C3000 this would be a great advantage, however, the same is also done for the C3100 and C3200 which may or may not be as advantageous. The C3100 and C3200 are more or less treated like a C3000 on OZ since they are almost identical as opposed to what other distros do, which treat the C3100 like a C1000 with an extra harddisk.

OpenZaurus also boasts to have around 6000 packages and that their package dependencies all work. However, this is not entirely true. They naturally forgot to mention how many of those packages actually work. Unfortunately, a large percentage of those packages don't actually work or are just mostly useless cannon fodder. Their large number also is a result that they split up every component into a separate package. So while some distros convenniently have one single package for an application, OZ would split it up into several dozen packages. Thus they need to heavily rely on their package dependency to pull all these packages together. Needless to say that not all their packages are up to date and not all dependencies resolve properly. Some packages have not been updated for quite some time.

I believe that OZ has a lot of potential to be the best distro eventually. See my custom OZ page to see a list of customisations that I have applied. However, in its current state, it can be summarised as fast but useless. OPIE looks good enough and is quite fast, but it lacks most of the essential applications like a decent browser and office applications and it is not backward compatible and thus cannot run applications from Sharp/Cacko. GPE is quite ugly but pretty fast as well. However, it is buggy and unstable too. In theory, OZ can do everything perfectly, however, in practice, it is quite a long way from that goal.

Zaurus users generally want to take advantage of all the features available on their superior devices and not lose functionality that is standard and working quite nicely with the stock Sharp ROM/distro. One thing the OZ team needs to do to woe more Zaurus users is to provide at least the same feature rich applications and functionality as the other distros already have. They should expand their repository to include more useful applications and actually test their packages before they put them on their feeds. They also need to get off their high horse and not just say that OZ is the best and everything else is crap. There are a lot of great functionality the other distros have that the OZ people could learn from and adopt to make it better. It is quite ironic that a distro that so heavily depends on package dependencies does not have a working GUI Package Manager. Both the GPE and OPIE package managers crash when installing large packages or attempt to handle package dependencies. The OZ flash utility/installer is also a joke as it is not even able to (re)partition the install target/disk which the other "inferior" distros can do. The keyboard mapping in OZ is less functional than Sharp/Cacko because it has much cleaner code and is not hacked together. However, I like these hacks (ie keyhelper) because they are actually very useful and make the Zaurus a pleasure to use.

OpenBSD

Apart from Linux distros, there is also OpenBSD for the Zaurus. Not sure whether I am going to try it though. Learning Linux is already keeping me quite busy.

Boot Loader

There are also alternate boot loaders for the Zaurus other than the default one provided by Sharp. With the introduction of the 2.6 kernel into the Zaurus world, several variants of boot loaders have been developed. pdaXrom has a new boot loader starting with beta4 called uboot. OZ has simplified the pivot boot process with altboot and once kexec is integrated, it could be used to boot other distros as well. There is also a special boot loader specifically for the C3000 featured on piro's website (http://www.piro.hopto.org/~piro/zaurus/bootloader/).

Using one of these bootloaders might be a way to boot and pivot the rootfs on the C3000 to overcome the tiny 16M flash problem.

OpenZaurus vs pdaXrom - a distro for the SL-C3000

I am quite happy with Sharp ROM (with a bit of tweaking) on my SL-C3100 since it behaves the way I like and does everything I want it to do. However, the SL-C3000 is a bit different since it got a tiny flash. Hence I am looking for an alternate distro for the SL-C3000 which can run completely off the internal MicroDrive.

The choice for a distro for my SL-C3000 is quite hard since none of the currently available distros are quite ready yet. Since my SL-C3100 is already running a Qtopia based distro, I really want a X11 based distro for the SL-C3000. There are two possible candidates, OZ/GPE and pdaXrom. However, neither of the two has gotten to a point yet where it is stable enough to use and includes sufficient useful applications and utilities. Both still have a lot of shortcomings.

Below is a discussion about why these two distros have potential and also where they fall short.

Firstly, the OS needs to be able to boot and mount the internal MicroDrive as rootfs. OZ does that very well, however, pdaXrom does not officially support the C3000 so it needs to be hacked to do that. Also, the perfect distro needs to fully support the SL-C3000 hardware and its many powerful features.

I will compare the two most relatively stable version, GPE based on OZ 3.5.4.1 (2.6 kernel) and pdaXrom 1.1.0 beta1 (2.4.20 kernel) rather than the most bleeding edge and experimental versions such as Angstrom GPE (2.6 kernel) and pdaXrom r121 or later (2.6 kernel).

Hardware Support

The SL-C3000 has great hardware which more or less is supported by both distros. Here is a quick summary of hardware and their level of support:

- XScale CPU - both OZ and pdaXrom are optimised for XScale
- CPU scaling and Power consumption - this is poorly supported on either distro
- touch screen - both distro support touch screen
- 4 GB internal hard drive - supported by OZ by default, ie OZ runs off the harddrive. pdaXrom requires hack to use harddrive but works just the same with hack
- USB host capability - OZ has support for USB host, but common device drivers are not automatically loaded. better supported by pdaXrom. USB storage is supported by both, however, usb mouse and keyboard is better supported by pdaXrom
- CF pcmcia service works on both OZ and pdaXrom, however, the CF card is not automatically mounted after reboot
- SD card service works on both but by default is significantly slower on OZ since it uses the **sync** option instead of **async**

- IrDA - seems to work on both distros but not much tested since it is not that useful these days
- Sound - is flakey on both pdaXrom and OZ. pdaXrom loses sound on suspend/resume while OZ does not have proper volume control and autosensing when the headphone is plugged in
- Rotation - between landscape and portrait style is supported by both distros, ie it is detected, but the resulting rotation is not so perfect depending on the window manager used. it works correctly on pdaXrom with openbox, but needs hack to work properly on OZ with matchbox

Keyboard Mapping

The main keys on the keyboard is mapped correctly on both distros, but is not very well documented on how to customise the keymaps. Some special keys are missing on both distros. Sticky keys can be enabled on pdaXrom which does not seem to be the case in OZ. Multi-key does not seem to work on OZ.

Window Manager

OZ uses the matchbox window manager which looks quite ugly. Fluxbox is also available. pdaXrom by default uses openbox with matchbox components as applets. It looks very nice. There are also a host of alternate window managers available for pdaXrom such as icewm, xfce, fluxbox and kde. All those window managers can automatically resize the application window on load to fit the screen except for openbox which is the pdaXrom default window manager.

TaskBar and Applets

The pdaXrom taskbar looks exceptionally pretty while OZ taskbar is rather dull. None of them have a docking view for minimised applications. A taskbar applet is available for both distros as well as a set of other useful applets such as clock applets and battery/cpu status applets. However, OZ does not have a working applet for ejecting removable cards. None of them have a swap management applet.

Themes

The default theme on pdaXrom is quite pretty and can be customised through the theme config applet, but switching themes requires manual editing of the config file. The OZ theme, on the other hand can easily be switched and customised through its config applet, however, switching the theme on OZ results in some applications, namely abiword, to crash.

Config Tools

The config tools in both pdaXrom and OZ are a bit flakey and don't always work properly. The date/time tool is very hard to invoke from the taskbar in pdaXrom, although it works when invoked from the menu. The date/time tool for OZ does not work. You cannot use it to change the date or time. There is no applet in OZ to configure sound. The pdaXrom sound applet has no effect while applications that use sound are active.

Suspend/Resume

Suspend and resume is a bit flakey on pdaXrom. When it works, it works fine, but at times, it does not work reliably. Suspend works fine in OZ, however, resume sometimes behaves flakey.

Screen Capture

Screen capturing is an easy task in OZ, however, this feature is missing in the default pdaXrom install. It can be added with contributed packages, although it is not as integrated as in OZ.

Applications

There are more useful applications in pdaXrom. There may be more packages in OZ, but the number of truely useful applications is greater on pdaXrom although some are oversized and don't fit on the Z's screen. A lot of the dialog boxes in both OZ and pdaXrom are also oversized. Some keys are incorrectly mapped in some of the applications. Most applications depending on SDL libraries, for example, have wrongly mapped keys. Firefox also has many missing keys on OZ.

Games

pdaXrom is the king in games. Not only can it support many games through emulators, but there are also a lot of user contributed games of all sorts. Even a playstation emulator has been ported and optimised for pdaXrom.

Language

The default language of both pdaXrom and OZ is English. Support for other languages is provided via way of unicode fonts and input method support for entering extended characters using a virtual keyboard. OZ supports a host of European languages, but lacks support for Asian languages such as CJK (Chinese/Japanese/Korean). pdaXrom has support for CJK, although not all applications have been compiled with CJK support enabled.

Development

Now this is the biggest difference between pdaXrom and OZ. OZ uses a build system called OE (OpenEmbedded) which is used to build everything in OZ. OE only runs on Linux and can target various platforms of which the Zaurus is one, but you will need a PC to build OZ packages and compile code. The native compiler which can compile code directly on the Zaurus itself is not very well supported on OZ. pdaXrom in contrast is more geared towards the Zaurus. It has a pre-made and configured development image that can be placed onto the Zaurus which allows you to immediately begin building and compiling. There are also bootable ISO images and VMWare images of pdaXrom for the PC, so you can easily run pdaXrom on your PC as well, or use the ready build image to cross compile applications for the Zaurus. OZ developers are also concentrating more on developing PDA specific applications that works on other devices besides the Zaurus, ie they target devices with smaller screens and less storage. pdaXrom on the other hand is more about getting as many PC linux apps working on the Zaurus as possible and exploiting and utilising every possible feature that the Zaurus has. In essence, pdaXrom is about turning the Zaurus into a mini linux PC, whereas OZ is more about providing a bleeding edge and alternate open source operating system, as well as lightweight applications for PDAs.

Support and Community

The OZ community is divided into two distinct groups, the developers and the end users. OZ developer prefer to communicate through their mailing list which is very development centric and offer support on IRC for users. They also announce things through their website and through posts on OESF. Documentation was almost non existant or scattered until recently. The new OZ wiki is a start for documenting a few things, but bug reporting is still quite confusing and tedious since there are several bug reporting systems and repositories.

pdaXrom has a much smaller development team than OZ but has a bigger user community (OZ has more users, but no cohesive user community). Support for pdaXrom is pretty much community driven. The pdaXrom users help each other and collect workarounds for common problems until they are fixed. Bug reporting is much easier and simpler in pdaXrom, but the versioning scheme of pdaXrom is quite confusing since it does not adhere to a standard.

Custom C3000 distro

Since neither pdaXrom nor OpenZaurus have sufficient features that would make them the distro of choice on a C3000, I have come to the conclusion that I would need to build my own custom distro for my C3000. This custom distro would be based on either pdaXrom or OpenZaurus.

pdaXrom development has stopped for the relatively stable and functional 2.4 kernel based beta1 and beta3. All current development is on the 2.6 kernel based beta4 and r121.

OpenZaurus is divided into OPIE and GPE. OPIE seems abandoned and does not offer much more than Cacko/Sharp in terms of functionality since it lacks major applications and is not backwards compatible with existing non-opensource applications. GPE seems unfinished and looks rather plain.

One option would be to continue work on pdaXrom beta3 and enhance its stability. The other option would be to enhance the Hentges OZ/GPE image. For the time being, I will refer to the stable pdaXrom beta1/beta3 based distro as **pdaXii13** (pdaXrom improved and integrated beta1 and beta3) and the OpenZaurus based distro as **OZii31** (OpenZaurus improved and internationalized 3.5.4.1).

Both of these possibilities would be X11 based.

I have decided to build pdaXii13 for my SL-C3000 which is based on the C3000 beta2 but much more polished. I have fixed all the shortcomings and bugs that annoyed me and now my C3000 has a real stable and perfectly working distro. See my pdaXii13 page on all the enhancements and features.

In the future I plan to work on pdaXiiNG which will be the next generation of pdaXii13 but based on the 2.6 kernel.

pdaXrom customisation:

The pdaXrom is a complete X windows environment for the Zaurus. It makes the C3000 and C3100 a true mini laptop.

Note: Development for 2.4.20 kernel based pdaXrom 1.1.0 beta1/beta3 has been discontinued by the developers in favour of a 2.6.x kernel based release.

The following section covers implementing workarounds for bugs and improvements to the 2.4.20 kernel based system. Most of these customisations should be generic for pdaXrom running on any Zaurus model. However, there are also a few C3000 specific customisations that should not be applied to the C3100 or any other models even though these two models are very similar hardware-wise. The major difference between the C3000 version of pdaXrom to other versions is that pdaXrom does not reside on the flash but rather is entirely located on the internal MicroDrive using a pivot boot feature.

Some of the customisations require additional files which should be downloaded first and placed on a SD or CF card for easy transfer to the Zaurus.

Fixing the boot process

There are several things during the boot process that can be fixed.

The first and most important one is the pivot boot feature that comes with the C3000 beta2 release of pdaXrom 1.1.0 beta1. The improved fix allows the C3000 to boot even if an external CF is inserted during the boot process and allows both the external and internal CFs to be mounted properly.

```
# cp init /media/realroot/sbin/init
# chmod 755 /media/realroot/sbin/init
# cp halt /etc/rc.d/init.d/halt
# chmod 755 /etc/rc.d/init.d/halt
# cp ide.opts /etc/pcmcia/ide.opts
```

It probably also is a good idea to run fsck before mounting the disk just in case. The suspend feature also seems to be happier on a clean disk. Often failure to suspend is corrected after the disk is fscked cleanly. The init script has been enhanced to do it if it can find fsck which you can copy to the initial root partition as follows:

```
# cd /media/realroot/lib
# cp /lib/libblkid.so.1.0 .
# ln -s libblkid.so.1.0 libblkid.so.1
# cp /lib/libuuid.so.1.2 .
# ln -s libuuid.so.1.2 libuuid.so.1
# cp /lib/libext2fs.so.2.4 .
# ln -s libext2fs.so.2.4 libext2fs.so.2
# cp /lib/libcom_err.so.2.1 .
# ln -s libcom_err.so.2.1 libcom_err.so.2
# cd /media/realroot/sbin
# cp /sbin/fsck .
# cp /sbin/e2fsck .
# ln -s e2fsck fsck.ext2
# ln -s e2fsck fsck.ext3
```

And finally, there are some minor fixes to eliminate and hide some annoying bootup error messages.

Hiding the *cardservices cannot find modules* message

```
# mkdir -p /media/realroot/lib/modules/2.4.20
```

Start X GUI automatically

Usually, runlevel 3 means booting into the console and runlevel 5 is for GUI mode. Since pdaXrom boots into runlevel 5, it should start X windows. The following fix will start X windows automatically during boot.

```
# head -`wc -l /etc/rc.d/rc|awk '{print $1-1}'` /etc/rc.d/rc > /etc/rc.d/rc.new
# echo "if [ \$runlevel -eq 5 ]; then" >> /etc/rc.d/rc.new
# echo " export HOME=/home/root" >> /etc/rc.d/rc.new
# echo " /usr/X11R6/bin/startx" >> /etc/rc.d/rc.new
# echo "fi" >> /etc/rc.d/rc.new
# echo "exit 0" >> /etc/rc.d/rc.new
# mv /etc/rc.d/rc.new /etc/rc.d/rc
```

The above has the same effect as adding the following towards the end of */etc/rc.d/rc*, but before the line that reads *exit 0* :

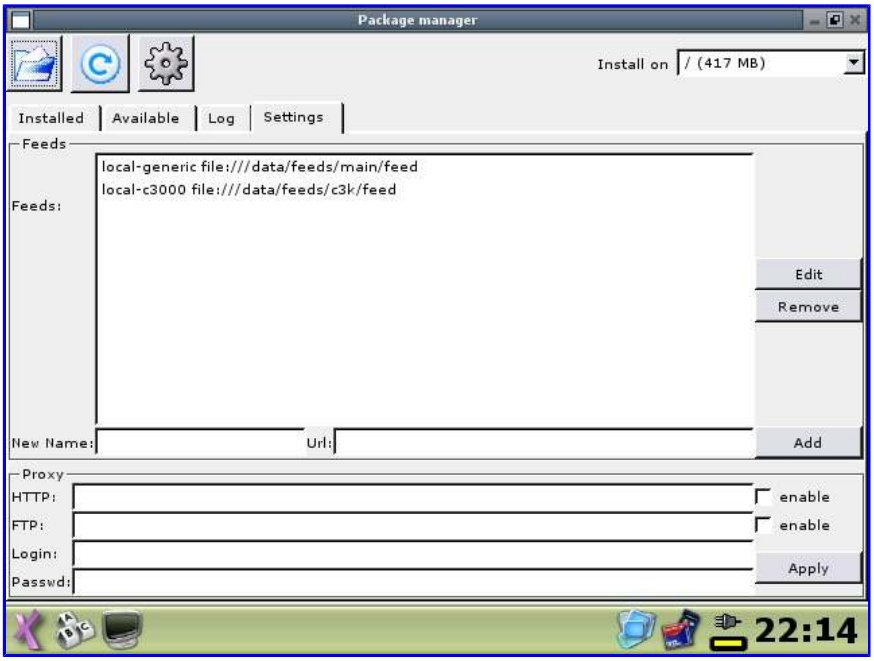
```
if [ $runlevel -eq 5 ]; then
export HOME=/home/root
/usr/X11R6/bin/startx
fi
```

Configuring Feeds and Installing Packages

The default pre-configured feed for the package manager is very out of date and needs to be replaced. There are two feeds for the C3000, a generic feed which is common to most Zauri models and a C3x00 specific one:

- http://mail.pdaxrom.org/download/1.1.0beta1/Zaurus-C1000-C3100/feed/
- http://mail.pdaxrom.org/download/1.1.0beta1/Zaurus-7x0-860/feed/

Alternatively, you can also just download both of the feed.tar.gz files and extract the content to a card or the internal disk. Then point the package manager at the directory containing the feed contents (eg: file:///home/root/Documents/installs/C3000/feeds).



You can also install individual packages without requiring a feed, but you will need to manually install dependant packages yourself. You can either use the GUI Package Manager or use the command line **ipkg** command.

For local installs, you can also merge the two feeds (and any additional ones) onto your local disk. Just extract the files into a directory (extract the 7x0 archive first and then the C3100 one on top of it to eliminate the duplicates). Then use **ipkg-make-index** to generate a new *Packages* file for the feed. I have included a copy of ipkg-make-index in my ipktools package [ipktools_0.2_armv5tel.ipk].

Additional Packages

There are many more packages beside the ones on the official feeds. Many user contributed packages exist and can also be installed. Below is a summary of additional packages that I installed from both the feeds and some were installed manually as stand alone packages:

required libraries:

- libstdcxx5-compat (needed for older applications)

useful command line utilities:

- vim (replace vi with vim because vi locks up all the time)
- lynx (great text based browser)
- aumix (volume controller)
- mplayer (video player)
- flite (speech synthesizer)
- bzip2 (compression utility)
- unrar (uncompression utility)
- unzip (uncompression utility)
- diffutils (text comparison utility)
- aspell (spelling engine)
- imagemagic (image conversion)
- kismet (wifi detector)

engines and runtimes:

- samba (file sharing)
- perl (interpretor)
- dosbox (DOS emulator)
- x11vnc (vnc server)
- java (classpath and jamvm)

games:

- doom2 (prboom doom engine)
- quake2 (quake2 engine)
- supertux

applications:

- gtkterm2 (tabbed terminal emulator)
- bluefish (HTML editor)
- tuxpaint (graphics)
- inkscape (graphics)
- xmmsplayer (video player plugin for xmms)
- gaim (IM client)
- tightvnc (vnc client)
- rox (file manager)
- emelfm2 (file manager)
- xarchiver (file compressor/decompressor GUI)
- gthumb (image viewer)
- gftp (ftp client)
- qbedic (dictionary)
- stardict (dictionary)
- imposter (presentation)

Some of the packages require further customisation and configuration before they can be used. Later sections of this guide show how those need to be configured.

Fixing the keymap

There are two types of changes to the keymappings. The first one are the essential fixes to make the keymap work, and the second ones are enhancements to make it more usable and similar to the keyhelper enhancements on the Sharp distro.

To customise or fix the default keymap, modify /etc/sysconfig/keyboard/kernel.map and /etc/X11/kb/akita.xmodmap and use *loadkeys* and *xmodmap* to reload the key mappings respectively. Use *xev* within X to determine the keycode for the keys.

You can also use the Input setup tool from the Systems Tools to remap the shortcut keys at the bottom of the keyboard. However, this seems to mess up the default keymapping, in particular, you will lose the . key so make sure you make a copy of kernel.map before you use the Input setup tool and then use loadkeys to reload kernel.map.

To enable sticky keys, you need to modify /usr/X11/bin/startx and replace **-kb** with **+kb +accessx** under the defaultserverargs variable. Also modify /etc/X11/kb/akita.xmodmap and uncomment the block of code around **keycode=73** except the last line. This will disable Tab to toggle Caps Lock. Now restart X and press the Shift key 5 times consecutively and sticky keys are enabled.

The keys are mapped as displayed on the keyboard except for some additional keys which have also been mapped but are not marked on the keyboard. Also some keys have been remapped as follows:

- Fn+q = Scandinavian A with loop
- Fn+o = Danish O crossed out
- Fn+p = British Pound sign
- Fn+a = AE character
- Fn+s = Euro symbol
- Fn+m = Backtick (accent grave)
- LeftKanji = Alt
- RightKanji = Backtick (same as Fn+m)
- Calendar = Control
- Address = Alt

I rewrote akita.xmodmap to mirror the Spitz/Borzoi keyboard according to the values returned by *xev*. The result is that each key on the keyboard is now clearly mapped and reflected in the new file that I called **spitz.xmodmap**. I also removed the duplicate mapping for Alt and Control from the Calendar and Address keys so that they can be reassigned, and also swapped Fn+m and Fn+q around. The right kanji key has also been remapped as a modifier key (Super L).

By defeault, there are also several predefined key combos which do the following:

- Ctrl+Alt+BS = Shutdown X
- Ctrl+Alt+Left = Switch to previous Virtual Desktop
- Ctrl+Alt+Right = Switch to next Virtual Desktop
- Shift+Alt+Left = Send to previous Desktop
- Shift+Alt+Right = Send to next Desktop
- Ctrl+Alt+d = Toggle Show Desktop
- Ctrl+Alt+m = Toggle Maximize/Restore Window
- Fn+Ctrl+Alt+Down = MoveRelativeVert
- Fn+Ctrl+Alt+Up = MoveRelativeVert
- Fn+Ctrl+Alt+Left = MoveRelativeHorz
- Fn+Ctrl+Alt+Right = MoveRelativeHorz
- Alt+Tab = Switch forward between running Applications
- Alt+Shift+Tab = Switch backward between running Applications
- Alt+0 = MaximizeFull
- Alt+5 = UnmaximizeFull
- Alt+9 = Iconify
- Alt+8 = Resize
- Alt+7 = Move
- Alt+4 = Close
- Menu = Activate Launch Menu

- Shift+Home = suspend
- Mail = sylpheed

I have assigned the Calendar and Address keys as follows:

- Calendar = aterm
- Address = xmms

Unfortunately, the following mapped keys did not work:

- Fn+1 = VGA mode (320x240)
- Fn+2 = SVGA mode (640x480)
- Fn+3 = Decrease Brightness
- Fn+4 = Increase Brightness
- Alt+F12 = ToggleShade

Hence I changed the mapping so that the Super key can be used instead of the Fn key. I also remapped Alt+F12 to Alt+6.

To reduce finger movements, I also added extra mappings so that instead of pressing Fn+Ctrl+Alt, you can also use the Home key instead, and instead of Ctrl+Alt, you can use the Super key (right kanji key). I also associated most Alt+key combos to be also activated by Home+key as well.

In addition, I have added the following shortcuts:

- Calendar = aterm
- Shift+Calendar = xterm
- Fn+Calendar = network config
- Ctrl+Calendar = usb config
- Alt+Calendar = ppp config
- Super+Calendar = installer
- Home+Calendar = dialer
- Address = xmms
- Shift+Address = keyboard
- Fn+Address = mixer
- Ctrl+Address = calculator
- Alt+Address = image viewer
- Super+Address = pdf viewer
- Home+Address = gimp
- Mail = sylpheed
- Shift+Mail = firefox
- Fn+Mail = thunderbird
- Ctrl+Mail = dillo
- Alt+Mail = xchat
- Super+Mail = gnumeric
- Home+Mail = abiword
- Ctrl+Alt+c = screencap
- Home+5 = Refresh Screen

The above changes can be done by modifying *rc.xml* which is by default located under */etc/xdg/openbox/*, however, it may also be under /home/root/.config/openbox/

For XMMS, the following keys can be used to control it:

- X = Play
- C = Pause
- V = Stop
- Up = Increase Volume
- Down = Decrease Volume
- Left = Skip Backward
- Right = Skip Forward
- Z = Prev Song
- B = Next Song

However, those keys aren't very convennient especially if you use the Zaurus in portrait mode and are using the buttons and scrollwheel at the back. Because of that, I modified xmms to be more Zaurus friendly so that you can use OK to play, Cancel to stop, and Up and Down to switch between tracks. I have reassigned N and M to be the volume keys.

You can either replace the **xmms** binary with my hacked version if you have already installed xmms, or install xmms_1.2.10-2_armv5tel.ipk instead.

xpdf has the following key mappings:

- o = open file
- r = reload
- f = find next
- n = next page
- p + previous page
- + = zoom in
- - = zoom out
- z = zoom page
- w = zoom page width
- 0 = zoom normal
- q = quit

You can either replace the **xpdf** binary with my hacked version if you have already installed xpdf, or install xpdf_3.01-2_armv5tel.ipk instead.

When you plug in a USB keyboard, you can automatically remap the keys so that the keys on the

external keyboard are mapped correctly, but it will result in the Zaurus keyboard being mismapped. When the external keyboard is unplugged, the keymapping for the zaurus keyboard is restored.

This can be done by creating a script in the */etc/hotplug* directory and catching the USB event to detect the insertion and removal of a USB keyboard and running **xmodmap** to load the appropriate keyboard map file.

Screen Resolution

The screen resolution can be switched between 640x480 and 320x240 using **xrandr** command. It can also be used to rotate the screen.

The *Home* + *1* and *Super* + *1* key combos should switch the screen to 320x240 if my custom keymapping is applied. Similarly, *Home* + *2* and *Super* + *2* will switch to 640x480 mode.

Screen Rotation

Screen rotation works on pdaXrom. Openbox will detect transformation between laptop mode (landscape) and PDA mode (portrait) and rotate the screen appropriately. You will end up with the taskbar at the correct location for the orientation you are in. However, the currently running applications are unaware of the rotation and their window size remain unchanged. I have written a script, **rotateall**, which rotates all the application windows to adjust them to the new orientation. To automatically invoke the script when the Zaurus is rotated, modify rc.xml (/etc/xdg/openbox/rc.xml) and replace *chkhinge* with *rotateall*. For best results, run the applications in full screen mode, ie maximised. I have written the script in such a way that if you run **rotateall** from a console without the screen actually rotated it will rotate the screen to the right instead of to the left. Running the script again will restore the previous orientation.

Screen Capture

A screenshot can be taken by running xwd -display :0 > screendump.xwd

A smarter script can be written that mimics the capture behaviour on the Sharp distro, ie bind the script to a key combo (eg Ctrl+Alt+c) and play a shutter sound before doing the screen dump. Also the filenaming and sequencing can also be made to mimic the Sharp snapshot tool. See the **screencap** script.

The screencap script I have written, also calls a .wav file (click.wav) to make a sound before taking the snapshots. If you install ImageMagick as well, then it will be able to convert the xwd image into png automatically.

Window Managers

pdaXrom uses openbox as its default window manager because it is lightweight and loads pretty fast.

However, there are many other alternate window managers (wm) that can be installed and used as well such as xfce, blackbox, fluxbox, matchbox, icewm, kde and enlightment. You can also setup multiple window managers and switch between them as long as you have sufficient space.

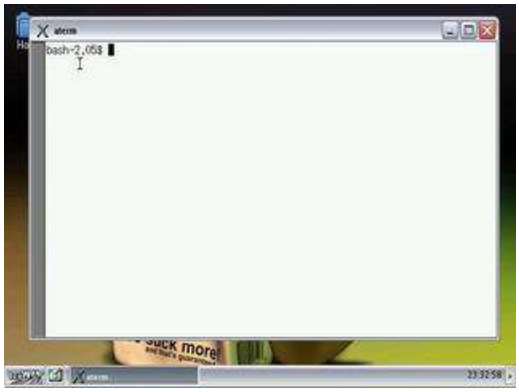
The default openbox window manager uses matchbox panels and applets for the taskbar and desktop management.

kde

KDE is also very popular, however, it is quite heavy and thus a bit slower. The simplest way of getting kde 3.5 installed is to use the kde squashfs image. Copy it to the internal harddrive and mount it as a loop device under /usr/lib/kde/kde/kde3.5. Then setup the symbolic links by installing the kdesymlink package.

Now modify .xinitrc and startx to run the selected default window manager. My customised scripts allow you to run either window managers depending on the WM variable, ie ob or kde. You can also start kde by running startxkde.

icewm



xfce

enlightment

Desktop and Menu Customisation

The .desktop files defining the applications are located under */usr/share/applications* and the icons are under */usr/share/pixmaps*

As mentioned before, the desktop is by default managed by matchbox desktop. You can either access the application icons via the menu or the desktop panels.

However, you can replace the matchbox desktop with rox filer to get an integrated file manager and desktop manager. To do that, modify */home/root/.xinitrc* and replace the following:

```
eval "matchbox-desktop $MBBGND" 2>/dev/null >/dev/null &
```

with this:

```
eval "rox --pinboard=MYPINBOARD" &
```

and then restart X.

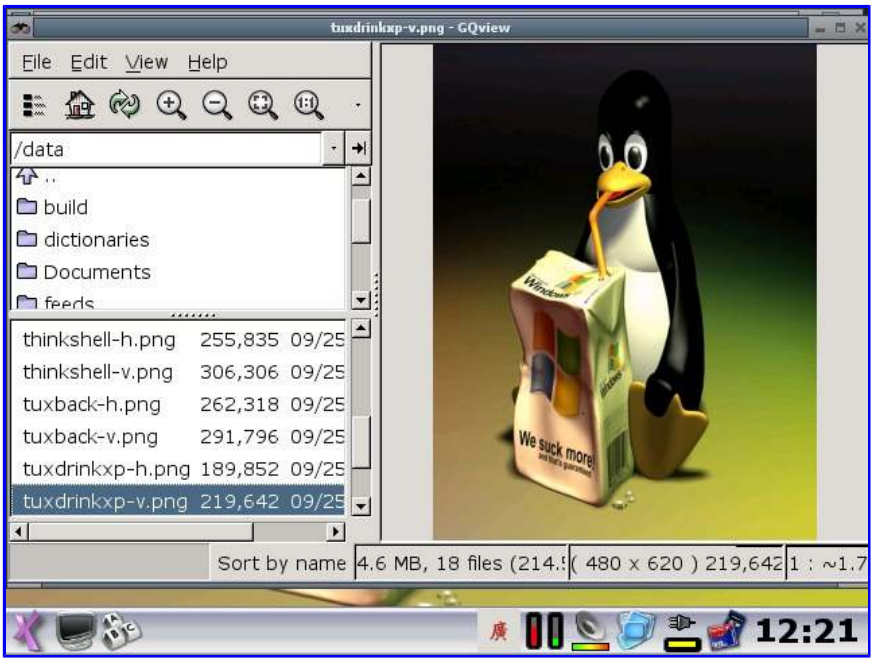
To change the theme, you need to edit rc.xml and modify the theme section. The themes are located under */usr/share/themes*. Each directory under there contains a theme. Place the name of the theme in the theme section inside the **name** tag. To apply the change, either restart X or run the following command:

```
# openbox --replace 2>/dev/null &
```

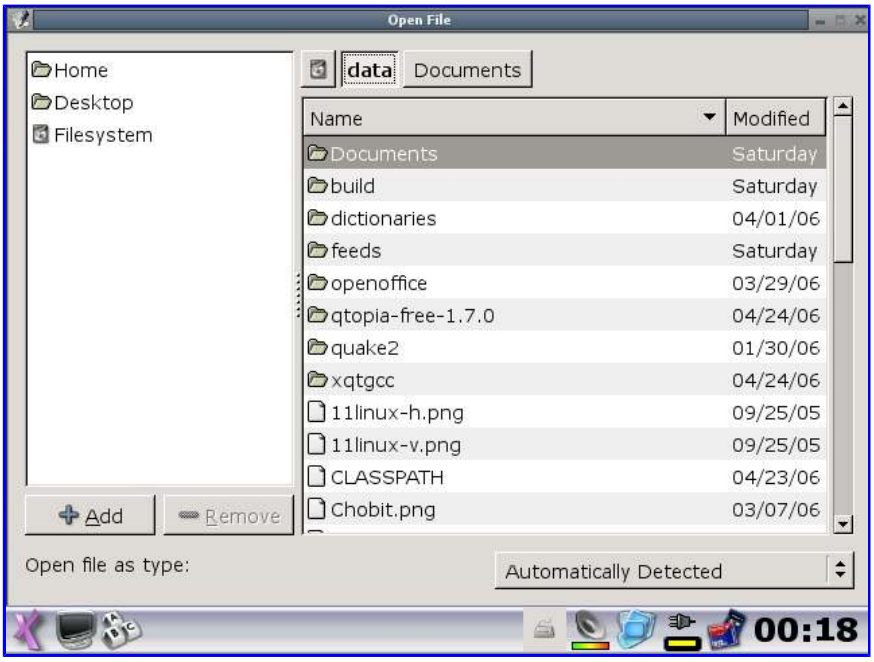
Window and Dialog Sizes

There are quite a few applications which open up in a window that is way too big for the Zaurus. This can usually be worked around by maximising the window in question (press *Home+m* or *Super+m*).

Some application can be tailored to a specific window size by editing their config files. Other applications have the size hardcoded. I have recompiled a few such applications, ie GQView and AbiWord.

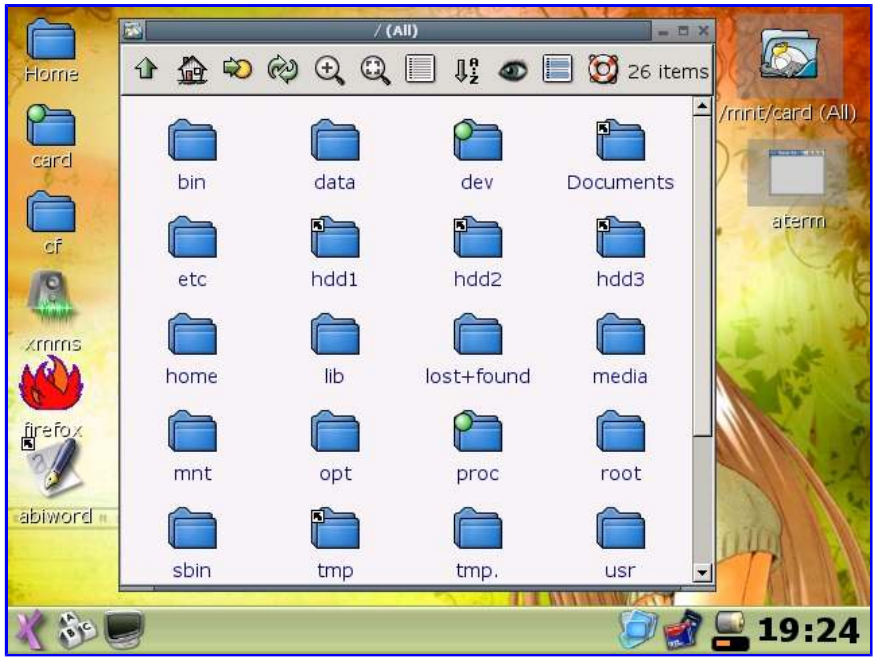


The File Open Dialog is also quite big and this problem is across most GTK2 applications. I have recompiled GTK2 and hacked the FileChooser to be more suitable for the Zaurus screen. You can either install my updated GTK2 package or replace libgtk-2.0.so.0.600.2 manually.

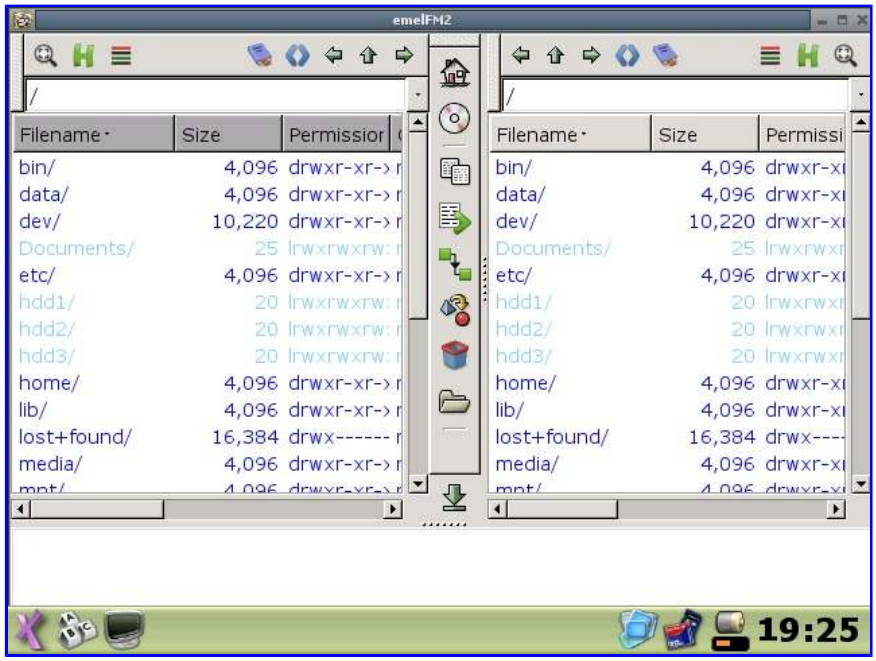


File Managers

There is no default file manager in pdaXrom, however, rox filer is a good candidate for this function. It has a similar interface to the konqueror for file management.

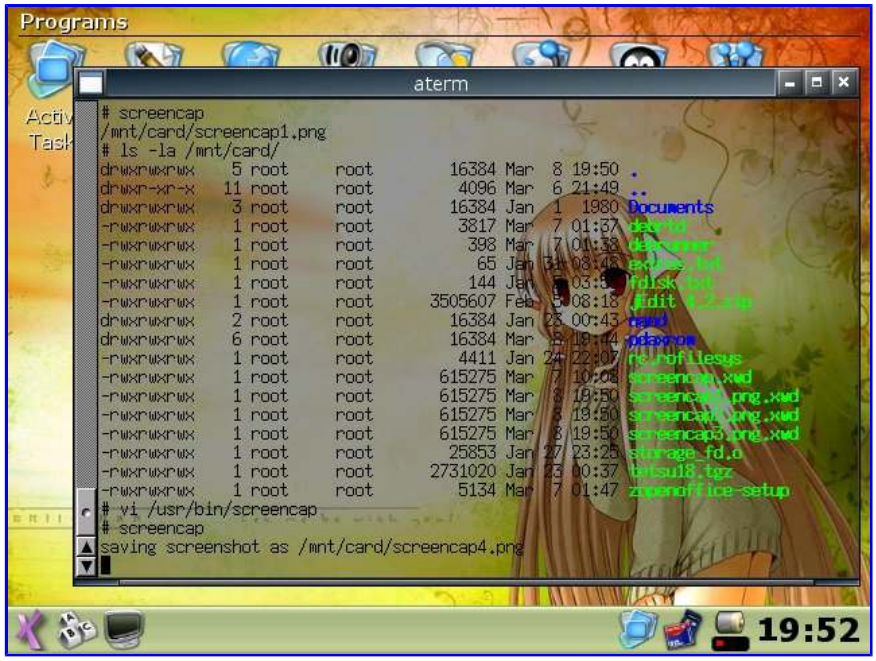


In addition, emelFM2 is also a good file manager with two panels.



Terminals

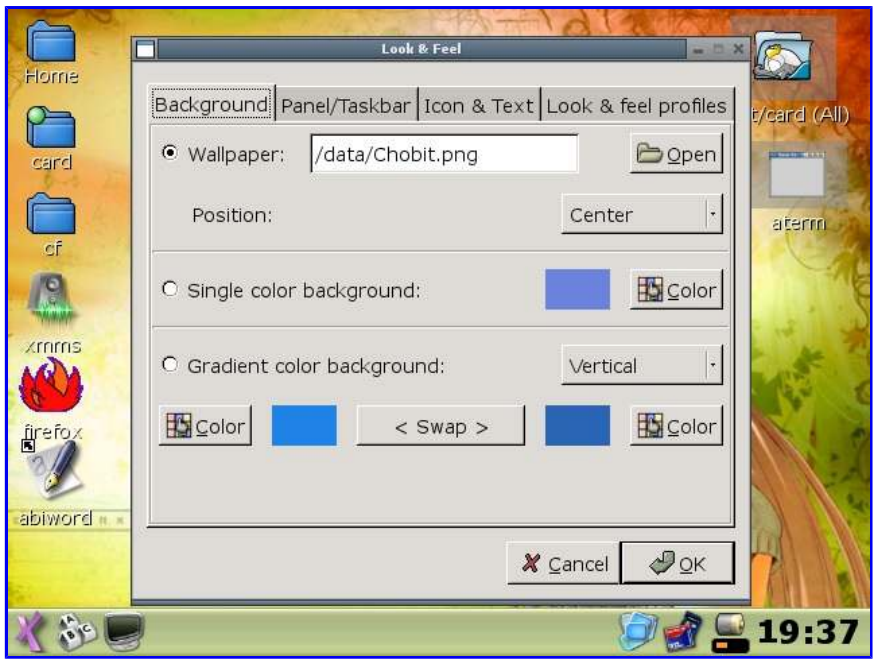
aterm is the default terminal in pdaXrom. It also supports transparent background. To enable transparency, copy Xdefaults to /home/root as .Xdefaults



Use rxvt as the terminal if you need to enter or view UTF8 encoded unicode characters.

Wallpaper Customisation

The Look & Feel tool in the Systems Tools can be used to change the default wallpaper as well as change many other aspects of how the GUI looks like. The wallpaper can be any png image file, but a resolution of 640x440 is ideal.



However, if you changed your desktop to use rox filer instead of the matchbox desktop, then you can right click (Fn+tap) to activate the rox filer properties where you can drag and drop an image into the background section.

ScreenSaver Customisation

For a X screensaver, install xscreensaver_4.24_armv5tel.ipk

zaurus user

pdaXrom by default does not have a zaurus user and thus runs everything as root. This would not be a problem if you don't connect to the net as the Zaurus is a single user environment and you are the only one using it. However, if you intend to connect your Zaurus to another computer or a network, or even the internet, then creating a zaurus user to run things as would be a good idea. You should also consider a firewall. To create a zaurus user, use the **adduser** command to add the zaurus user and then set the suid bit on busybox so the zaurus user can su and do other things.

Resume and Suspend

The suspend and resume is a bit flakey. Sometimes, the hard disk light does not turn off, but a subsequent resume and suspend will make it suspend with lights off. fscking the disk will often allow clean suspends.

Also, when the Z is suspended while xmms is active, then the sound device is locked until a reboot. Please ensure that you add esdctl suspend into the apm suspend script and have esdctl resume in the corresponding resume script to automatically suspend and resume the sound daemon.

MMC driver

There is a new OpenSource SD/MMC driver for the C3100/C1000. In order to test it on the C3000 as well, install mmc-driver_2.4.20_arm.ipk. This driver works, but currently does not automount nor does it unmount cleanly from the taskbar. Use **/etc/sdcontrol** [insert|eject] to mount and unmount the SD card.

Alternatively, a fully functional SD driver that supports 2GB+ cards has been made by Sharp and released with the SL-C3200. This driver [sdmmc-module_2.4.20_arm.ipk] also works on pdaXrom (beta1) because it is using the same kernel as the Sharp ROM, ie 2.4.20

auto mounting

Automatic mounting via /etc/fstab does not work in pdaXrom. Also, there are only two loop devices by default just like Sharp ROM. I have created an automnt script to automaticaly create additional loop devices and mount cramfs and squashfs images as well as enabling swap. I have linked /etc/rc.d/init.d/automnt to /etc/rc.d/rc5.d/S90mount and /etc/rc.d/rc6.d/K30unmount

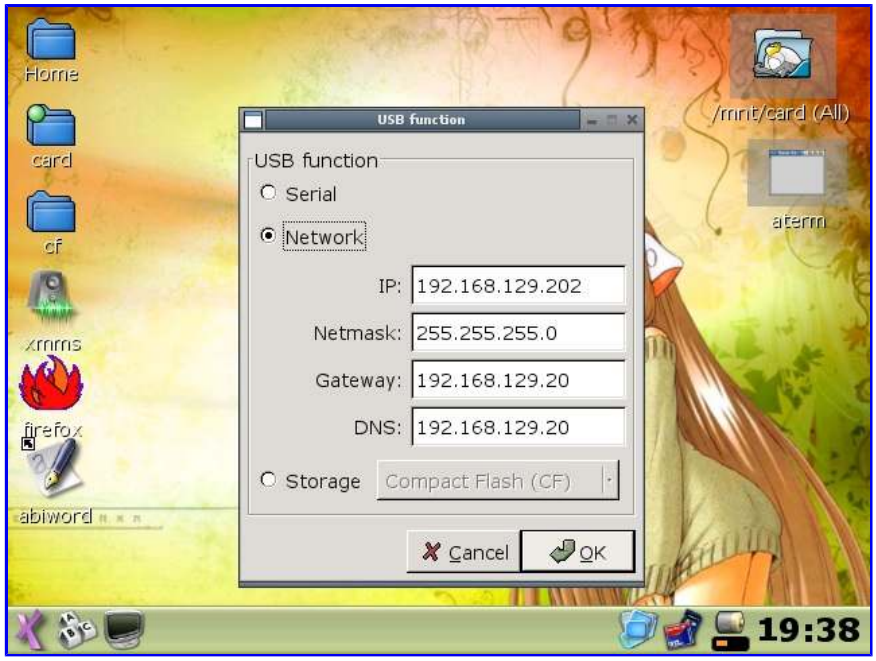
ftp and telnet

The C3000 version of pdaXrom does not have telnet nor ftp pre-installed. The version of busybox that comes with it does not include telnet and the binary on the pre-pivot root system is compiled with a newer version of glibc and thus won't run. I have copied the binaries for ftp and telnet from the Sharp distro and they work if placed in the right location. Since telnet really is busybox, I had to rename the busybox binary to telnet and it needs to be placed into /bin. I placed ftp into /usr/bin.

Alternatively, you can also install inetutils-clients_1.4.2_armv5tel.ipk from the unstable feed as well.

USB network

The USB network module works in two modes similar to the Sharp distro. It can either allow the Zaurus to share a disk or used as a virtual network adaptor. The Zaurus can be mounted as a USB storage device when connected with the USB sync cable. Copy the updated **usbfunction.py** to /usr/bin which will allow you to either share the SD, CF or internal MicroDrive. Sharing the internal MicroDrive might be a bit tricky with a Windows machine since it is formatted as ext3.



Alternatively, the USB network mode will allow you to connect the Zaurus via a virtual ethernet network and thus will allow you to share files via Samba if configured and also allow you to use any TCP/IP dependant network services if enabled. If for example the PC or Laptop you connect to via USB cable has internet sharing (Windows ICS or iptables on Linux) configured, then the Zaurus can access the internet through that connection. (see the networking section of my customisation guide for Sharp ROM on how to do that. I have also written a script **usbnet** which automatically configures the usbd0 interface to use the Windows ICS (Internet Connection Sharing).

Samba

You need to install the Samba package and configure */etc/samba/smb.conf* to share the desired shares. It is also a good idea to create a user such as zaurus and assign a password to that user using **adduser** -H. Then enable and create the zaurus user for samba using **smbpasswd** -L -a -e options. Then restart samba service using **/etc/rc.d/init.d/samba** stop and start as arguments.

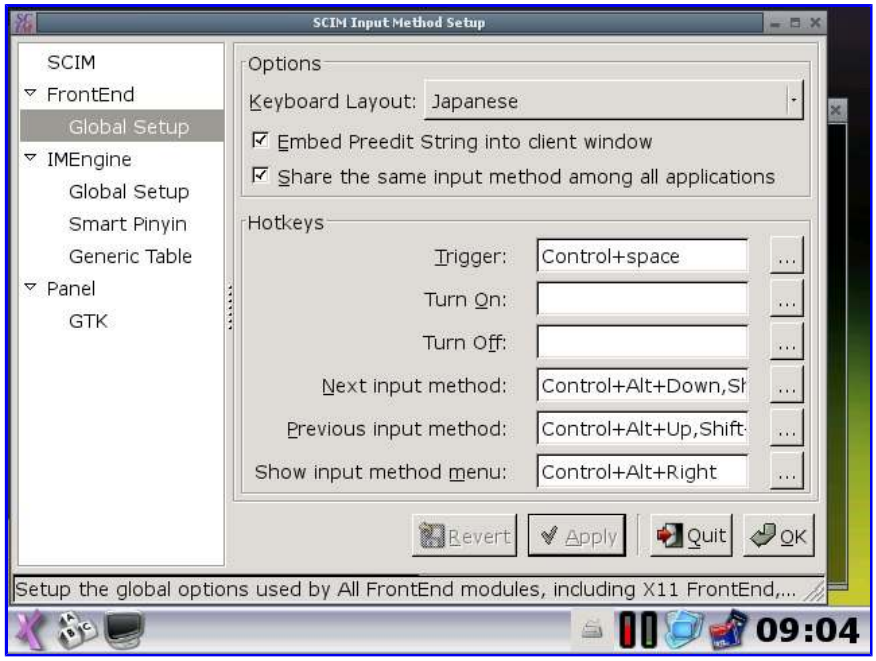
You can manually start and stop the samba service by running the script /etc/rc.d/init.d/samba which gets run at bootup. This script is very chatty and displays too many errors. I have modified it to be less chatty and also added a restart option.

If you enable the USB network, then you need to restart samba so it can pickup and listen on the new interface. You can also automate that by modifying /etc/hotplug/usbd.func and force samba to restart after the USB networking module is loaded.

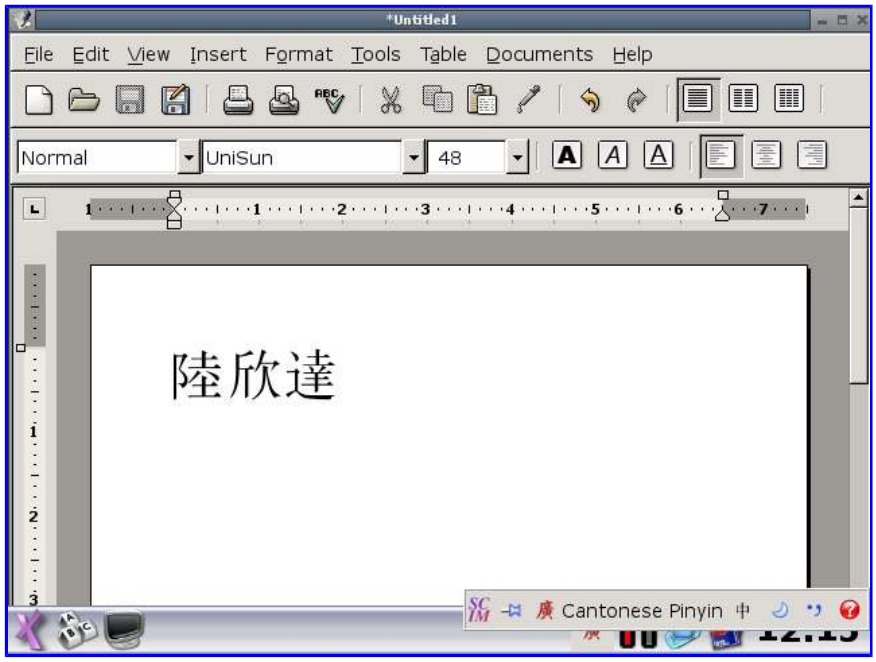
Enabling Chinese and Japanese

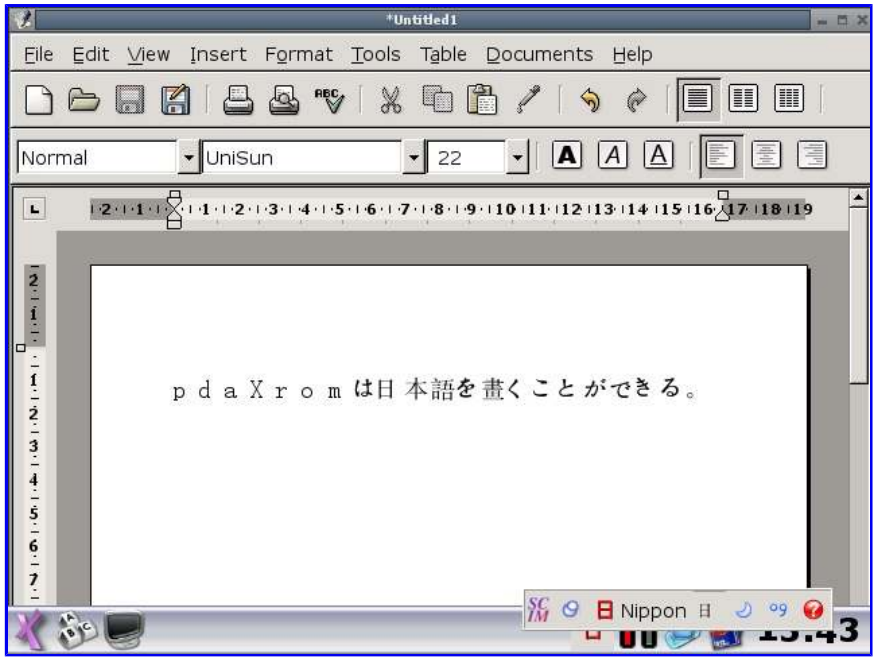
Chinese and Japanese menus can be enabled on pdaXrom by installing a Chinese or Japanese locale and installing Chinese and Japanese fonts. TTF fonts go into /usr/X11/lib/X11/fonts/TTF and pcf fonts go into /usr/X11/lib/X11/fonts/misc. You also need to install an input method engine for entering Chinese and Japanese characters. I installed scim, scim-pinyin and scim-tables for that.

Once scim is installed, you can configure it by running **scim-setup**.



Once scim is configured, you can run **scim -d** to start the scim engine and frontend as a daemon. Now for applications that scim can hook into, you can activate and switch input methods by pressing *Ctrl+space*. AbiWord for example can be used with scim:

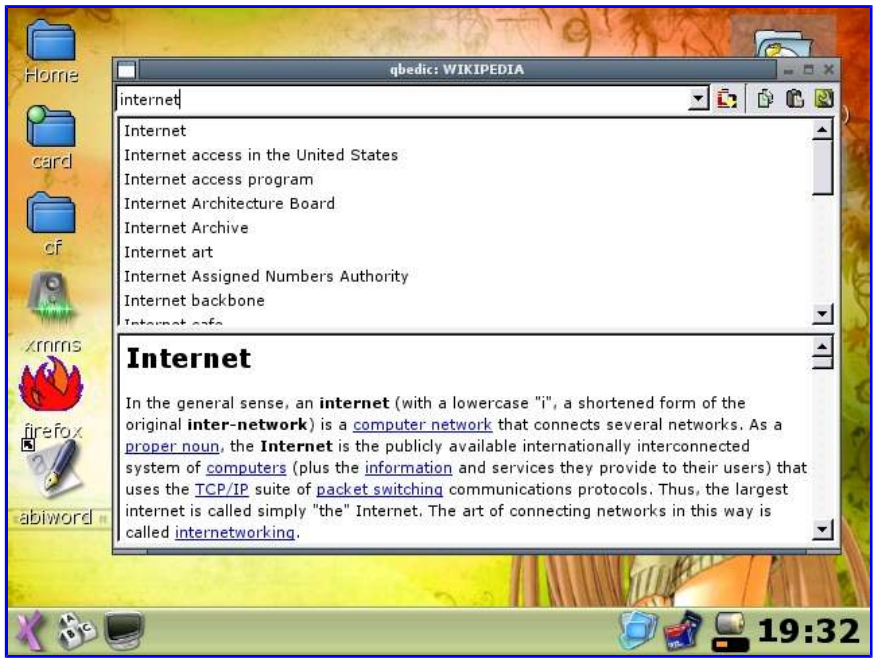




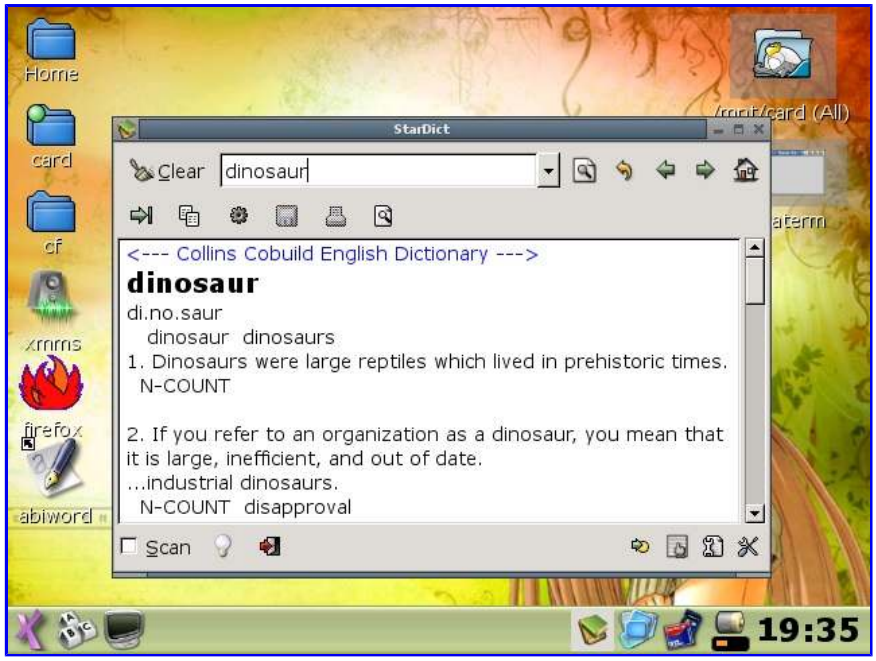
Remember to start scim each time you start X, or load scim from xinitrc to start scim automatically.

Dictionaries

There are several dictionary packages available for pdaXrom. QBEdic is a popular dictionary between different languages that is also available for the Sharp ROM (called ZBEDict), so the same dictionary files can be used. There is also a complete wikipedia that can be installed into QBEDic for reference. QBEDic dictionary files can be placed into any directory, but by default they are located under /usr/share/bedic. You can configure QBEDic to look in specific directories for dictionary files.


Home
card
cf
xmms
firefox
abiword
qbedic: WIKIPEDIA
internet
Internet
Internet access in the United States
Internet access program
Internet Architecture Board
Internet Archive
Internet art
Internet Assigned Numbers Authority
Internet backbone
Internet
In the general sense, an internet (with a lowercase "i", a shortened form of the original inter-network) is a computer network that connects several networks. As a proper noun, the Internet is the publicly available internationally interconnected system of computers (plus the information and services they provide to their users) that uses the TCP/IP suite of packet switching communications protocols. Thus, the largest internet is called simply "the" Internet. The art of connecting networks in this way is called internetworking.
19:32

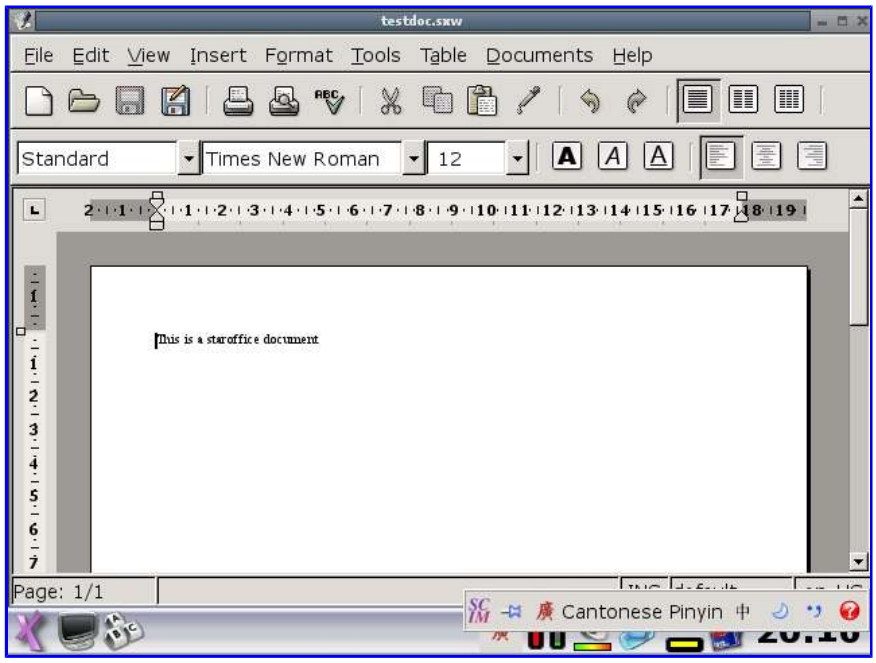
Another popular dictionary package for pdaXrom is stardict which also runs on Windows. There are many dictionary files available for stardict, and there is also a sound package for stardict to pronounce the words in English. Stardict expects its dictionary files to be located under /usr/share/stardict/dic. The sound files have to go under /usr/share



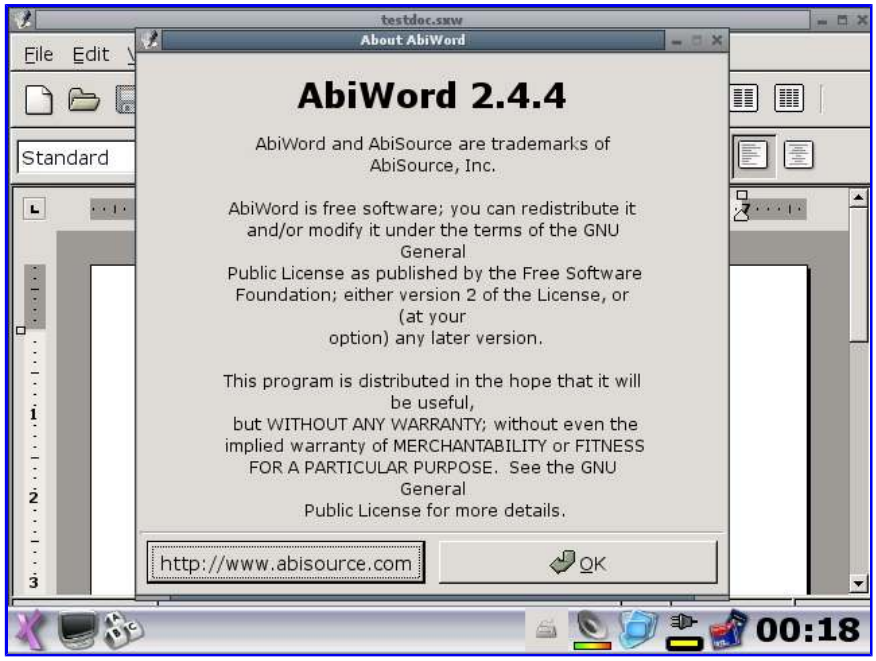
I have created a squashfs image containing several qbedic and stardict dictionaries as well as the sound files. Since the dictionary files are quite huge, placing them into a squashfs image can save quite a lot of space. I have also created a script called linkdics.sh which created the necessary links for stardict so that the files on the squashfs image can be located by stardict.

AbiWord

Abiword is a nice word processor, but it defaults to a 800x600 screen resolution which is not very usable for a Zaurus. For the C3000, it should be changed to 640x480. This can be done by editing */home/root/.AbiSuite/AbiWord.Profile*



I have compiled Abiword 2.4.4 and fixed all the window sizes so now every window in Abiword is sized correctly. I also compiled some text import/export plugins, so now I can open and edit StarOffice/OpenOffice documents.

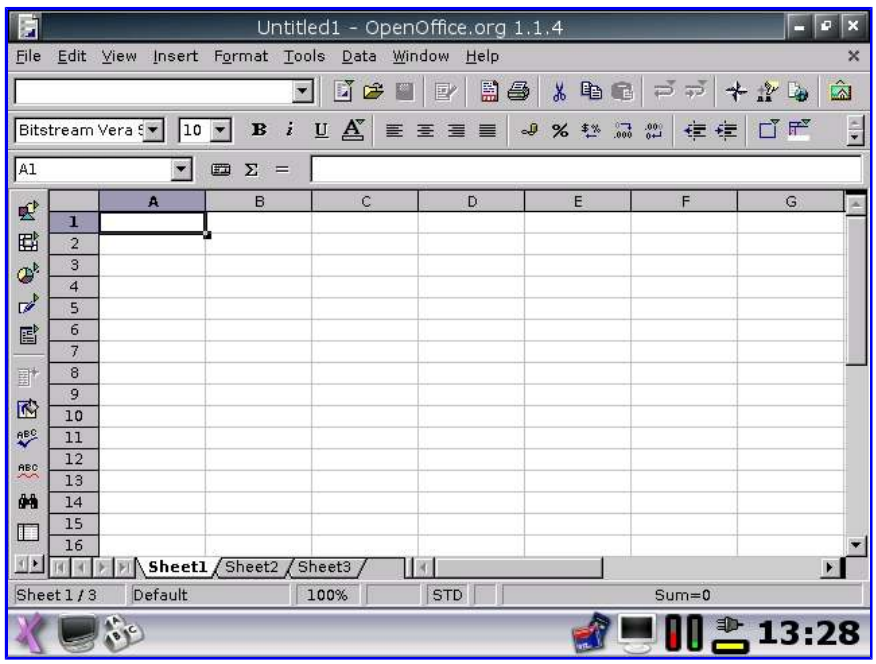

testdoc.sxw
File Edit
About AbiWord
AbiWord 2.4.4
AbiWord and AbiSource are trademarks of AbiSource, Inc.
AbiWord is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
http://www.abisource.com
OK
Standard
00:18

Installing aspell will enable spell checking in abiword.

OpenOffice

The Debian arm port of OpenOffice 1.1.4 works on the Zaurus. However, it is compiled against an experimental version of glibc 2.3 whereas most the Zaurus distros use a stable 2.2.x version of glibc. The usual way around this is to install PocketWorkstation (Debian arm) in chroot mode and install OpenOffice under PocketWorkstation. This certainly is inconvennient and quite a waste of space if you only want to run OpenOffice and none of the other Debian apps that come with PocketWorkstation since you would need at least 512MB of space for the two.

As an alternative, I have created a pre-installed image of OpenOffice on a cramfs as well as squashfs compressed image file which contains a bundled minimalistic version of Debian (minideb) and a pre-installed image of OpenOffice 1.1.4 binaries that can be run off the compressed image.



To install it, copy the cramfs or squashfs file to the internal harddisk or SD/CF card and mount the cramfs/squashfs file as a loop device. Then run the zopenoffice-setup script contained in the mounted image.

I have also included a tgz file, so if you have an ext2/ext3 formatted disk, you can extract the file and

run zopenoffice-setup from the location you extracted the files to.

In addition, I also created openoffice-cn.squashfs which is preconfigured for usage on Chinese Zauri and includes Chinese fonts.

Lastly, you need to modify */usr/X11R6/bin/startx* and remove "-nolisten tcp" from the defaultserverargs and restart X. You should see a new OpenOffice icon in the menu. Alternatively, you can also start it from the command line by issuing **runoo** in a terminal.

Quake and Quake2

Quake as well as QuakeII plays pretty well and smoothly on the Z with pdaXrom, but don't change the resolution or display settings from within Quake/QuakeII or you will get a white screen and nothing else will happen anymore.

If you own Quake2, then you can copy the entire baseq2 directory to /home/root/.quake2

mplayer

The pdaXrom feed for C3x00 contains a mplayer package which you can install and it works quite well. Some people claim that the mplayer package from Cacko which was compiled by AGAWA Koji (atty) is much faster because it is optimised to use bvdd and iwmmx [mplayer-bvdd-iwmmxt_1.1.5-1_arm.ipk]. However, analysing the mplayer package on the C3x00 feed [mplayer_1.0pre7try2_armv5tel.ipk] it is in fact a slightly newer package based on atty's latest code and compiled with the same options, ie bvdd and iwmmx.

I have repackaged [mplayer-bvdd-iwmmxt_1.1.5-1_arm.ipk] as [mplayer-bvdd-iwmmxt_1.1.5-1_armv5tel.ipk] and changed the file structure to reflect the pdaXrom file structure and moved things to /usr/share instead of /opt/QtPalmtop and I also renamed the mplayer binary to mplayer2 so you can install both versions and compare speed.

The following mplayer command line will allow you to watch video:

```
# mplayer -vo bvdd -ac mad -ao ss -framedrop -really-quiet file
```

To force fullscreen playback, add the -vm option:

```
# mplayer -vo bvdd -ac mad -ao ss -framedrop -really-quiet -vm file
```

The xmmsmplayer addon to xmms can be used as a frontend for mplayer.

It uses the -vo x11 option by default to play the video in a gtk window. You can make it use the bvdd driver instead and play fullscreen also. The config file used by the plugin is */usr/share/mplayer/mplayer.conf* and contains the following config:

```
framedrop = yes
cache = 1024
dr = yes
af=resample=44100
```

Add the following to make mplayer use the bvdd driver and play fullscreen:

```
vo = bvdd
vm = yes
```

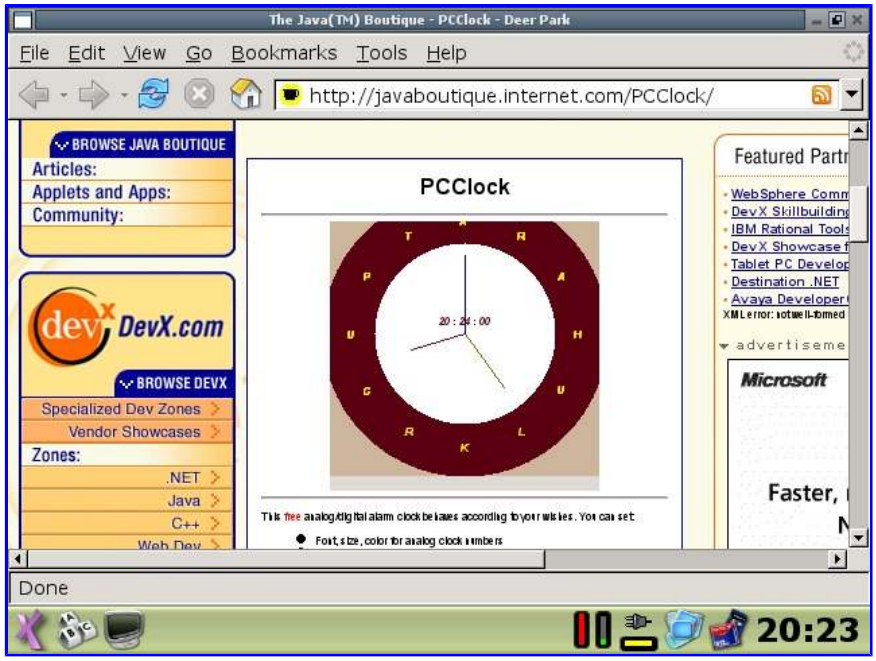
There is a bug with mplayer that it does not refresh the screen properly once it exits from fullscreen mode. As a workaround, you can run **xrandr -o normal** to refresh the screen. Alternatively, if you are using my customised rc.xml file for openbox, then you can simply press *Home+5* to refresh the screen.

I have also compiled the latest version of mplayer, 1.0pre8 with gmplayer enabled.

Java

Java is available through the jamvm and classpath packages. Using these packages, a JRE 1.4.x compatible environment can be achieved, however, due to copyright and licensing restrictions, jamvm cannot be called Java nor does it guarantee full compatability although it can run most Java applications. Using jamvm and classpath, I was able to run jedit on my Zaurus.

In addition, I managed to build a Java Plugin for Firefox which uses jamvm and classpath. Just install gcjwebplugin-0.3.2_armv5tel.ipk and Firefox 1.5 will be java enabled. It will be able to run most applets, however, not all applets will work. The plugin uses the classpath libraries so watch out for updates of the classpath package which should allow more applets to work.



The jamvm and classpath packages gives you a Java compatable runtime, however, in order to develop Java applications, you will need a Java compiler and other tools like jar, javah, javap and javadoc. Jikes can be used as a replacement Java compiler, and there is also a classpath tools package which has some of the Java tools you need.

If you want to be very sneaky, then you can copy the tools.jar file from a 1.4.x JDK and use it instead.

gcc

There is an on-board gcc 3.4.5 compiler to allow native development on the Zaurus. Furthermore, there are two different gcc images, one that generate soft float binaries (pdaXrom native) and one that generates hard float binaries (cross compile apps for Sharp/Cacko).

To use the gcc compiler, you first need to install the following packages that it requires:

- pcre
- xml-parser
- perl

Next put **zgcc-3.4.5.img** under */home/root*. This will make it being automatically mounted on boot. You can also mount it manually as follows:

```
# mount -o loop /home/root/zgcc-3.4.5.img /opt/native/arm/3.4.5-xscale-softvfp
```

To compile a little test application, create a file called hello.c with the following content:

```
#include <stdio.h>
int main()
{
printf("Hello World\n");
}
```

Then run the following command which will generate an executable called **hello**.

```
# gcc -o hello hello.c
```

To compile source packages, first download and extract the source tarball. Then do the following:

```
# ./configure
# ./make
```

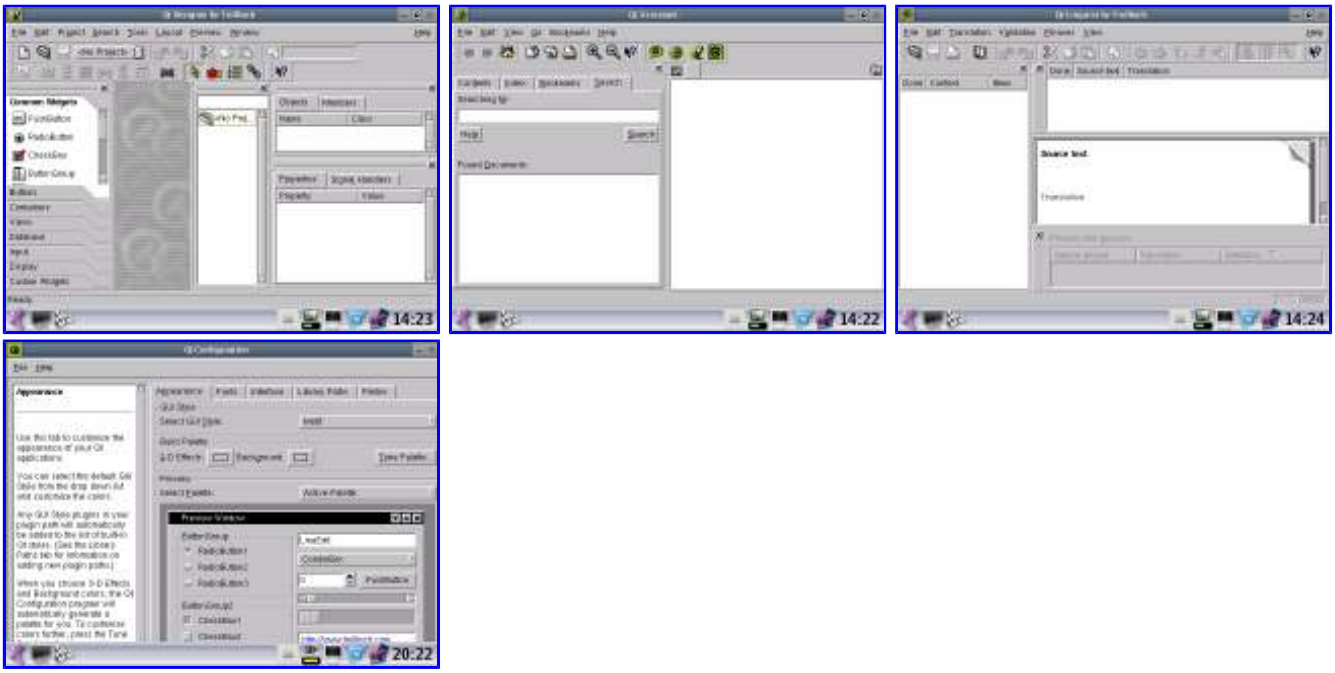
You can also use **strip** to strip the binary to make it smaller once it is built. Usually, there is a DESTDIR variable defined in the Makefile which can help you turn the compiled binaries into a package.

```
# cd /data/build/pkg
# newipk testapp
# cd /data/build/src/testapp
# make DESTDIR=/data/build/pkg/testapp/data install
# cd /data/build/pkg
# makeipk testapp
```

This example uses ipk-tools to create a package. See the ipkg section below for further info.

You can use gcc 3.4.5 to compile both console and X based applications, however, you cannot use it to compile the kernel or glibc. You will need gcc 2.95-2 instead.

I have slightly modified the gcc image and added a few missing symlinks and also added some binaries and scripts required for compiling some gnome packages. In addition, I also re-compressed it as a squashfs image rather than cramfs.



Furthermore, I also added QT designer, assistant and linguists, and I recompiled qtconfig so it won't startup oversized.

Building your own Packages

You can also build your own packages (ipk files) if you have written some useful scripts or written some applications that you want to distribute and let others install easily with the standard package manager.

I have build a package ipktools [ipktools_0.2_armv5tel.ipk] which has a set of tools for manipulating ipk files on pdaXrom:

- newipk - creates a package template structure for you to add files to for packaging
- makeipk - package up a directory that contains files in an ipk structure into an ipk file
- unpackipk - extracts the contents of an ipk file into a directory structure for repackaging
- deb2ipk - converts a deb file into an ipk file format (needs perl)
- zipipk - zip up an ipk file and remove ipk file afterwards
- ipkg-make-index - creates the Packages file for a feed

There currently are two ipk file formats. One uses the tar and gz format, wheras the other one uses a different binary format that is the same as the Debian .deb format. The Zaurus with default Sharp ROM (and Cacko) as well as pdaXrom uses the tar and gz format, which basically is a gzipped tarball (.tgz or .tar.gz) with a control structure and renamed to .ipk. If you extract this ipk file, you will find 3 files inside it - a text file called debian-binaries which just contains the string *2.0*, and two .tar.gz files called control and data. The control.tar.gz file contains a text file called control which has informaton about the package such as the Maintainer's name, dependencies, version, description, etc. There may also be some optional shell scripts for doing some pre and post configuration tasks during install and uninstall. Finally, the file data.tar.gz contains all the files and directory structure of the files for their destination location. The other ipk format is used by OpenZaurus and PocketWorkstation (Debian) and requires the **ar** command to extract the files.

To unpack an ipk file to see what is inside it, do the following:

```
# unpackipk somefile.ipk
```

To create your own ipk file, do the following to create the ipk file structure:

```
# newipk myproject
```

Then once you add your files in the correct locations and also update the control file with the information about your application, you can create your ipk file with the following command:

```
# makeipk myproject
```

If you want to convert a .deb or OZ ipk file so you can unpack it with **unpackipk**, then convert it using the **deb2ipk** perl script first, or ue the **ar** tool instead.

Zaurus Backup

You should always backup your system since that is the only way to recover if something goes wrong. The easiest way to backup is to archive everything using **tar** and then use **gzip** to compress the archive. I have also created a script which does that.

OpenZaurus customisation:

OpenZaurus 3.5.4.1 is an alternate Linux distribution for the outdated and proprietary bundle of Qtopia that is shipped with the Sharp Zaurus. OpenZaurus is based on open-source code and enables the Zaurus to run most of the current non-proprietary and open-source Linux applications. It is based on newer versions of Qt/Embedded (Qtopia) and also supports X11. OZ also brings an up to date 2.6 kernel to the Zaurus world instead of the old 2.4.20 kernel that Sharp shipped with the C3x00 series.

However, OZ comes in several differently packaged distributions or flavours. OZ/OPIE is the Qtopia based distribution wheras OZ/GPE is the X11 based distro. There is even an OZ/bootstrap with a bare minimum base system without GUI.

In general, OZ is faster than the legacy Sharp ROM or its derivatives such as Cacko. However, OZ lacks an artistic touch. The default theme in GPE is rather ugly, in particular the icons are very plain and the font sizes are badly picked. The default OPIE theme on the other hand looks much more polished with nicer icons and crystal clear fonts. However, though the icons for the OPIE desktop are nice, the same can not be said for the applications. The tiny icons designed for the 200x320 screen resolution look very rugged and out of place in the otherwise good looking OPIE desktop. Hopefully someone with graphical skills will come up with a nicer set of icons for OZ.

Featurewise, OZ has most of what Sharp ROM offers in terms of device support. Screen rotation, screen capture and resume/suspend works almost perfectly with only a few hickups. Softwarewise, OPIE has most of the software available on the Sharp ROM except for the proprietary software bundles. Unfortunately, those are quite important applications, in particular NetFront. GPE has most of the applications available for pdaXrom and PocketWorkstation, so a combo of OPIE+GPE can be comparable to a dual boot Cacko/pdaXrom or Cacko/PocketWorkstation type system.

Hentges is another OZ based distribution which is tailored to be more user friendly and has the most commonly used applications pre-installed. Hentges is also pre-configured to allow you to have both OPIE and GPE running at the same time (not yet released) so you don't need to choose between the two and end up being limited to the applications the choosen flavour provides. Running OPIE and GPE at the same time allows you to use all of the applications available to OZ.

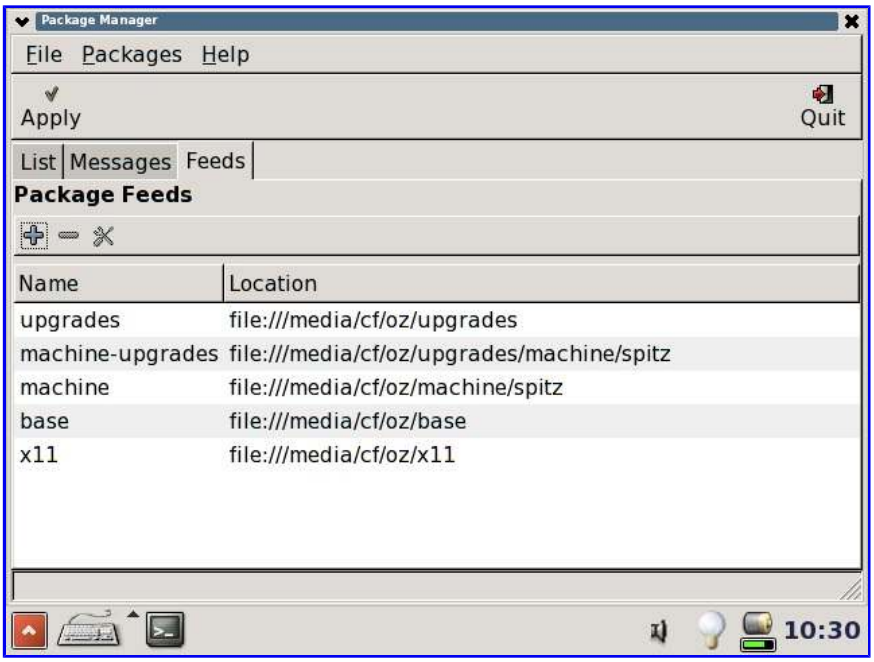
The following section covers how to best use the available packages in OpenZaurus. It also shows how to run both OPIE and GPE applications together to get the best of both the Qt/Embedded and X11 world.

Some of the customisations require additional files which should be downloaded first and placed on a SD or CF card for easy transfer to the Zaurus.

Configuring feeds

By default, the feeds are already configured to access the main feed server for installations. The Package Manager is either pre-configured for either OPIE or GPE, depending on which you installed, but if you want to run both concurrently, then you need to update the feed configuration to include all the feeds.

You can also change the feeds to point to one of the alternate mirror sites, or if you have mirrored the feeds to a CF or SD card, then you can also change the feed to download from your local file source as well which is quite handy if you are offline a lot.



The above sample shows a configuration with the OZ feeds mirrored to a CF card.

OPIE Package Manager uses /etc/ipkg.conf to configure all its feeds, whereas GPE Package Manager has individual files for each feed located under /etc/ipkg/

You should add all the additional feeds as well for a bigger choice of available applications for installation. OZ 3.5.4.1 can use packages from both the 3.5.4 and 3.5.4.1 feeds. There are quite a few application packages in the 3.5.4 feeds but not in the 3.5.4.1 feeds.

- base
- machine
- upgrades
- opie
- x11

- perl
- python
- contrib

After changing the config files, you will need to run ipkg update to refresh the configuration and package db.

```
# ipkg update
```

You will also find that you will get more success when using the command line **ipkg** command to install packages rather than the GUI Package Manager.

Installing OPIE with GPE

You can install both OPIE and GPE on different virtual terminals and switch between those two VTs to switch between OPIE and GPE. However, running both environments like this concurrently also uses up more resources. Alternatively, you can also pick one of the two environments to be your main environment and run the other in a shrink down version so you can just run the applications.

Personally, I believe OPIE is more mature than GPE as a windowing environment for the Zaurus, and the configuration items in OPIE also appear to be more relevant for the Zaurus than they are in GPE. Thus it makes more sense to add GPE functionality to OPIE rather than the other way around.

The following will assume that OPIE is installed as the main windowing environment...

Installing X/Qt under OPIE

Alternatively, you can also use X/Qt under OPIE to run X based applications without the need to switch virtual terminals. This feature would be similar to the X/Qt functionality under Sharp/Cacko. Unfortunately, unlike the X/Qt packages that are for Sharp/Cacko which has dedicated feeds that contain packages with basic default configurations, the X/Qt package under OPIE is a standalone package of the binary without any configuration scripts or whatsoever. In order to make use of it, quite a lot of manual customisation and tweaking is required under OPIE....

Installing GPE

Although GPE is quite rudimentary you still might want to run a full X server comparable in features and usability to pdaXrom. Using the Hentges GPE image would be a good start. It is much more usable than the standard GPE install. However, there is still a need for a considerate amount of customisation effort to be put into it....

Essential Stuff

There are several essential customisations in order to get OZ working properly.

Timezone

The default OZ install only has a few timezones predefined. If your timezone is not one of them, then you need to first install a package for your timezone and then configure it. My timezone package that I needed to add was: *timezones-australia*. Once you installed the timezone package, do the following according to your timezone:

```
# ln -s /usr/share/zoneinfo/Australia/Sydney /etc/localtime
```

The above is for the Australia/Sydney timezone. Substitute that part with the appropriate entry for you. Additionally, you also need to add **export TZ=Australia/Sydney** (substitute with appropriate value) to

hwclock.sh under /etc/init.d

You also need to modify **locale.conf** and change the timezone entry for OPIE.

Keymapping

The default keymapping is defined in /etc/keymap-2.6.map

You can customise this file to remap certain keys. The most common change is to exchange **,** with **/** which is done by modifying keycode 51, ie:

```
keycode 51 = slash comma
```

The ` key (grave) seems to be missing. I have assigned it to Fn + q:

```
altgr keycode 16 = grave
```

Also since on Sharp/Cacko ROM a common mapping for { and } is Fn + o and Fn + p, I have added mappings to mimick those familiar key mappings:

```
altgr keycode 24 = braceright
altgr keycode 25 = braceleft
```

Since Fn + d shows a . (dot) on the physical keyboard but nothing is mapped for it, I added a mapping:

```
altgr keycode 32 = period
```

I also remapped Fn + k back to the yen character since that is what is shown on the physical keyboard:

```
altgr keycode 37 = yen
```

Keymappings common to both GPE and OPIE need to be done in keymap-2.6.map, however, there are also keymappings specific for GPE and OPIE respectively which need to be configured separately. The GPE version of Hentges has fluxbox which uses /home/zaurus/.fluxbox/keys

GPE fixes

The default window manager in GPE is matchbox which runs all applications in maximised mode. In order to switch between several applications easily, it is a good idea to install the **tasks** applet and add it to the taskbar at the bottom of the screen.

It is also a good idea to install the **cards** applet as well, so you get to control the SD and CF cards.

matchbox also supports screen rotation, however, it is misconfigured in GPE. In order to fix it, add **--no-flip** to the **matchbox-panel** command in the matchbox session file:

```
matchbox-panel --orientation south --no-flip &
```

OPIE fixes

OPIE is similar to the Sharp distro/Cacko and a lot of the customisations for those can also be applied to OPIE since it is also based on Qtopia but is a newer version (version 2.3 vs version 1.5) and open source.

Unfortunately, the cardmon applet in OPIE is broken. It allows you to unmount the internal MicroDrive

and does not show the SD card. To fix this, uninstall or disable the pcmcia applet and install the cardmon applet from the 3.5.4 feed instead of the 3.5.4.1 feed.

You also need to install volumeapplet in order to be able to turn on/off keyboard clicks just like in Sharp/Cacko.

Additional Packages

OpenZaurus has a lot of packages in its official feed, however, it does not have everything pre-installed so many packages need to be manually installed once OZ has been installed. The following are the recommended ones that I added straight away:

Command Line tools and runtimes

- kernel-module-rtl8150
- kernel-module-squashfs
- kernel-module-usbkbd
- opie-confedit 1.2.1
- opie-embeddedkonsole 1.2.1
- opie-keypebble 1.2.1
- opie-volumeapplet 1.2.1
- opie-cardmon 1.2.1 (from 3.5.4 feed)
- sudo 1.6.8
- file 4.13
- diffutils 2.8.1
- gzip 1.3.5
- inetutils 1.4.2
- lsof 4.75
- x11vnc 0.7.1
- bitchx 1.1
- dosbox 0.65
- flite 1.2
- links 2.1
- mplayer 1.0pre8
- mencoder 1.0pre8
- unzip 5.52
- zip 2.32
- jamvm 1.4.2
- classpath 0.20
- jikes 1.21

Applications

- mc 4.6.1
- mrxvt 0.4.1
- abiword 2.4.5
- abiword plugins: eml, hancom, jpeg, mswrite, opendocument, openwriter, pdf, sdw, wml, xhtml
- firefox 1.0.6
- sylpheed 2.2.4
- gftp 2.0.18
- gputty 0.9.8
- epdfview 0.1.2
- qpdf 2.10
- gphoto2 2.1.6
- gqview 2.1.1
- gxine 0.4.1
- prboom 2.2.6

The following are applications that I really need to get working. These either don't exist in OpenZaurus yet, or are out of date and/or need fixing:

- OpenOffice
- fbvnc (for Spitz)
- tightvnc

- firefox 1.5 with plugins (java,flash,mplayer)
- putty
- emelfm2
- xarchiver
- qbedic
- stardict
- gthumb
- imagemagick
- gimp
- festival
- xmms (with Zaurus friendly keys)
- quake2
- dukenukem3d

Using Packages from other distros

OpenZaurus can also run some packages from other distros. This depends on the dependencies of those packages. For example, OPIE can run some of the Sharp/Cacko binaries because they are also based on Qt/E. OPIE is based on a newer version of Qt/E than Cacko and Sharp ROM and can run quite a few of those applications using compat libraries. However, some Sharp/Cacko applications depend on proprietary non-opensource libraries which cannot be run on OPIE. Opera for example can be run on OPIE.

Some pdaXrom packages can also be run on GPE since both are based on X11 and GTK+. However, pdaXrom also has many applications that use other display widgets such as QT or fltk which GPE does not have by default. Also, pdaXrom packages before beta4 depend on a separate iconv library which is not part of glibc. Packages compiled after beta4 no longer have this dependency just like OZ packages and also use a similar kernel as well and thus should be even more compatible.

Nevertheless, I have tried a few pdaXrom beta1 packages on OZ and they work fine. So instead of trying to build useful packages that don't exist in OZ, I just installed the pdaXrom package on OZ instead. The following are some of the packages that just work:

pdaXrom packages:

- multi-aterm
- putty
- jikes

Similarly, debian arm packages (PocketWorkstation) can also be installed on OZ. The .deb file can be easily converted to .ipk files. In fact, the OZ .ipk format is almost identical to the .deb format.

USB Networking (via client cable)

The virtual USB network interface via the USB client cable does not currently work with Windows. It does however work with Linux. This is just standard **ifconfig** and **route** stuff once you have loaded the gadget module. You will need to load it manually after plugging in the USB cable. Don't expect it to be loaded automatically like in Sharp/Cacko or pdaXrom.

Setting up WiFi

Just plug the CF WiFi card in and provided it is supported, the appropriate drivers will be loaded for it and you can configure your network using the network applet. This is straightforward stuff if you know how to setup a network.

Setting up Bluetooth

Just plug the CF Bluetooth card in and provided it is supported, the appropriate drivers will be loaded for

it and you can setup PAN or GPRS with the applet. Make sure you don't forget to set your PIN.

Setting up Ethernet network

Using a USB Ethernet adaptor you can connect to an ethernet network. Usually, those adaptors use a rtl8150 or pegasus chip so installing the appropriate kernel module will enable the adaptor as eth0 once plugged in. The rest is standard **ifconfig** and **route** stuff.

Configuring Samba and SSH

You can share the files on your Zaurus via Samba. Just install the samba packages and configure smb.conf. Furthermore, you can also get a remote terminal into the Zaurus via ssh and transfer files via scp. OZ by default uses dropbear instead of the more common sshd.

Configuring VNC server and client

Configuring VNC allows you to remotely control other machines or allow other machines control your Zaurus.

USB Keyboard and Mouse

You can plug in external USB keyboard and USB mouse if you have a USB host cable. Using a USB hub, you can have multiple USB devices connected concurrently. The USB keyboard should be correctly mapped and just work. The USB mouse might be a bit more problematic since it might fight for control with the stylus.

Chinese/Japanese support

Support for displaying and inputting Chinese and Japanese is possible. You will need to install unicode fonts with the extended character set that supports the extra Chinese and Japanese characters. Additionally, an input server such as SCIM and inputmethod plugins also need to be compiled and installed...

Backup

Backups are always important for recovery in case something goes wrong or accidental deletions. However, doing a NAND backup on any of the C3x00 series models in OZ 3.5.4.1 is useless since it does not even use the NAND flash area. The backup application provided by OZ only backs up the PIM data which is a good thing but not sufficient. A gzipped tarball of everything seems to be the only solution at the moment.

Multi Booting

OZ is able to facilitate booting of different media through the use of altboot. With the addition of kexec to altboot, it is even able to boot another kernel. Using these features you can boot different versions of OZ or even other distros from SD or CF cards.

Altboot allows you to boot straight off an ext2 filesystem from a physical media such as an inserted CF or SD card, or mount an ext2 loop back filesystem from either CF or SD card as well. I have enhanced the altboot config and menu to also allow you to boot off ext2 loop back images from the internal MicroDrive located on /hdd3 partition. In order to do that, place the loop back images under */media/hdd/boot-images* and add **30-bootHDD** to */etc/altboot-menu* and update *altboot.conf* and

altboot.func.

If you want to disable the altboot menu from being shown several times, then disable altboot on the ext2 loop image. This prevents altboot to be run a second time during bootup once you have selected which image to boot.

C/C++ development

A native on-board gcc compiler can be used to develop and compile applications directly on the Zaurus. Although this is not as fast as developing on a PC, it is quite useful when you are using your Zaurus as a laptop replacement and want to build your own apps without access to a PC running Linux.

To install gcc so you can develop X applications, do the following from a terminal:

```
# ipkg -nodeps install make diffutils pkgconfig
# ipkg install gcc-symlinks g++-symlinks binutils-symlinks libc6-dev gtk+-dev ncurses-dev libpng-dev libjpeg-dev libungif-dev libx11-dev libxpm-dev libxft-dev libxrender-dev libxt-dev xproto-dev libz-dev
```

Alternatively, I have also created a cramfs image containing a ready made gcc bundle similar to the pdaXrom native gcc image. The native gcc packages for OpenZaurus are based on gcc 3.4.4 and glibc 2.3. I have created an image zgcc-344.cramfs which was build using the armv5te packages from the 3.5.4.1 feed. The following packages are a minimum set that is required to compile C/C++ code and build simple packages:

- cpp_3.4.4-r5_armv5te.ipk
- cpp-symlinks_3.4.4-r5_armv5te.ipk
- binutils-dev_2.15.94.0.1-r1_arm.ipk
- binutils-symlinks_2.15.94.0.1-r1_arm.ipk
- binutils_2.15.94.0.1-r1_arm.ipk
- diffutils_2.8.1-r0_arm.ipk
- g++-symlinks_3.4.4-r5_arm.ipk
- g++_3.4.4-r5_arm.ipk
- gcc-symlinks_3.4.4-r5_armv5te.ipk
- gcc_3.4.4-r5_armv5te.ipk
- libc6-dev_2.3.5+cvs20050627-r3_armv5te.ipk
- libstdc++-dev_3.4.4-r5_arm.ipk
- make_3.80-r0_arm.ipk
- pkgconfig_0.15.0-r3_arm.ipk

With this mimimal set of packages, I can compile simple C/C++ programs. I have build a cramfs image (zgcc-344.cramfs) which includes many more development packages and tools for easy compilation of X11 applications similar to what the pdaXrom zgcc image offers. See packages.txt for a list of packages included in this cramfs image.

In addition to installing the packages, I also had to add a few symbolic links as well as apply some changes in the directory structure as well as some configuration changes. All the required changes have been captured in **mkzgcc.sh**.

To use the cramfs image, mount it under /mnt/zgcc and run the zgcc-setup script to create the necessary links. Finally, source zgcc-env and you are ready to compile.

```
# mount -o loop zgcc-344.cramfs /mnt/zgcc
# /mnt/zgcc/zgcc-setup
# . /mnt/zgcc/zgcc-env
```

To compile a little test application using the C compiler (gcc), create a file called hello.c with the following content:

```
#include <stdio.h>
int main()
```

```
{
printf("Hello World for C compiler\n");
}
```

Then run the following command which will generate an executable called **hello**.

```
# gcc -o hello hello.c
```

To compile a little test application using the C++ compiler (g++), create a file called hello.cpp with the following content:

```
#include <iostream>
using namespace std;
int main()
{
cout << "Hello World for C++ compiler\n";
}
```

Then run the following command which will generate an executable called **hello++**.

```
# g++ -o hello++ hello.cpp
```

To compile source packages, first download and extract the source tarball. Then do the following (for example grisbi):

```
# bzip2 -dc grisbi-0.5.9.tar.bz2 | tar xvf -
# cd grisbi-0.5.9
# ./configure
# ./make
# ./make install
```

Java development

A java compatible environment exists on the Zaurus via the use of jamvm and classpath as an alternate JVM and jikes as a compiler.

To install java so you can run and compile java applications, do the following from a terminal:

```
# ipkg install classpath
# ipkg -nodeps install libffi3 jamvm jikes
```

---

pdaXii13 for C3000:

pdaXii13 is a bug fixed and customised version of pdaXrom beta3 which uses the 2.4.20 kernel. pdaXii13 stands for pdaX integrated and improved beta1 merged with beta3. pdaXii13 was originally designed and build for the SL-C3000 but can now also be installed on similar Zaurii like the SL-C3100/SL-C3200 and SL-C1000. The SL-C3000 specific version is called pdaXii13 Spitz (Alice) while there also is pdaXii13 Akita (Sally) for the other models. The installation instructions on this page is focused on the SL-C3000, but there are other sections that describe installation for other models as well. Most of the customisations

done for pdaXii13 can also be applied to all the other clamshell Zaurii models since they are quite similar in many aspects, but there are also differences in some areas which is covered later.

pdaXii13 is based on pdaXrom which is a complete X windows environment for the Zaurus. It makes the SL-C3000 a true mini laptop. However, pdaXrom never officially supported the SL-C3000 due to its small flash size. pdaXrom beta3 was the last relatively stable release of pdaXrom based on the 2.4.20 kernel, however, it was never finished. The original pdaXii13 was the SL-C3000 specific version of pdaXrom beta3 with some extra finishing touches. Newer versions of pdaXii13 now also supports the NAND based sister models of the SL-C3000, namely the SL-C3100/SL-C3200 and SL-C1000. Both versions, pdaXii13 Spitz (Alice) and pdaXii13 Akita (Sally) are collectively known as simply pdaXii13 while pdaXii13 Spitz comes in a base and full edition, and pdaXii13 Akita only comes as a base edition.

The SL-C3000 was the first Zaurus with an internal MicroDrive released by Sharp. It came with only 16MB of NAND which is a major shortcoming. The later models in this series do not have this problem anymore and all have 128MB of NAND instead.

Because of that, the pdaXrom version for the SL-C3000 runs completely off the internal MicroDrive using a feature called pivot root which was originally developed for OpenZaurus. The pdaXii13 Spitz (Allice) edition of pdaXii13 does exactly that, ie runs completely off the internal MicroDrive. The pdaXii13 Akita (Sally) edition of pdaXii13 runs off the internal flash memory (NAND) the same way that pdaXrom beta3 does.

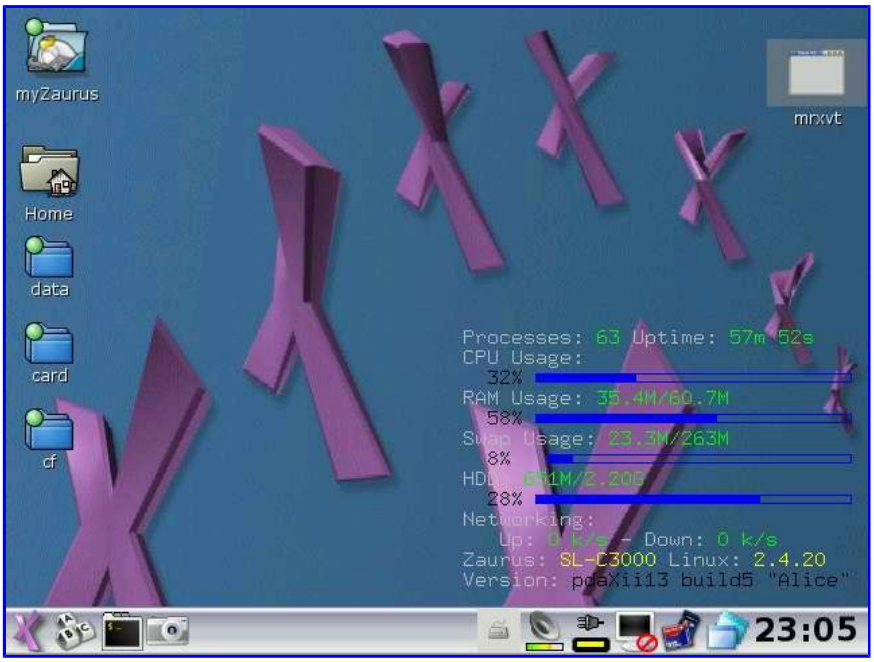
Enhancements

pdaXii13 is based on the pdaXrom C3000 beta2 release which was originally based on pdaXrom beta1 but was never updated after pdaXrom beta3 was released to address issues in beta1. This distro aims to update the pdaXrom beta1 based C3000 beta2 release to the pdaXrom beta3 release.

Additionally, pdaXii13 also includes the new SD card module to support SD cards larger than 2GB. It also allow the SL-C3000 to be detected as a USB storage device and share the third partition on the MicroDrive just like in Sharp/Cacko.

Enhancements and fixes discussed in my pdaXrom customisation section have also been rolled into pdaXii13 release as the default.

Once you have installed pdaXii13, you will have the applications listed further below installed and customised for the Zaurus. The customisations that have been applied to pdaXii13 differs from the default pdaXrom beta3 release. There are quite a lot of enhancements made in **pdaXii13 full** install. The **pdaXii13 base** install is pretty much like a bug fixed beta3 for SL-C3100 and SL-C1000 without much customisations nor additional packages installed. Unless explicitly mentioned, everything that follows is about pdaXii13 full which is heavily customised with many additionally pre-installed applications.



Installation

To flash pdaXii13 Spitz (Alice), you will first need to repartition your internal MicroDrive. The original partitioning provided by Sharp is quite adequate for the base install, however, if you had installed the previous C3000 betas, then you probably had your entire MicroDrive partitioned into a single 4GB filesystem. This is OK too. The first partition should be at least 512MB if you install the full pdaXii13, but pdaXii13 base should fit into the default Sharp partition layout. My personal preferred partitioning is as follows since I do a lot of native compilations on the Zaurus:

```
hdd1 2.2GB / (ext3/linux ID 83)
hdd2 256MB /swap (linux swap ID 82)
hdd3 1.5B /data (win95 fat ID c)
```

However, for most users, a larger data partition would probably be more useful and hence the newer pdaXii13 installer will create the following custom partition layout instead:

```
hdd1 1.4GB / (ext3/linux ID 83)
hdd2 256MB /swap (linux swap ID 82)
hdd3 2.4B /data (win95 fat ID c)
```

The pdaXii13 installer also allows you to create a single partition for the entire MicroDrive, or use an improved version of fdisk which can create partitions properly unlike the default fdisk which comes with the Zaurus to manually create your own desired partition scheme. There is also an option provided which allows you to restore your original factory set partition layout.

Before you flash to a new distro, you will first need to prepare the flash medium. For this you will need either a 512MB or 1GB SD/CF card formatted as fat16. Place all the required installation binaries onto the root of the card. The following files are required:

- updater.sh
- updater-tools.bin
- zImage-2.4.20.bin
- initrd.bin
- hdimage-full.tgz (see note below)

Note: You can either use hdimage-base.tgz which gives you a vanilla pdaXrom install with minimal/essential fixes only, hdimage-full.tgz which is a fully customised and polished image with all the customisations as described below, or hdimage-custom.tgz which is a snapshot/backup of a custom system. The installer will pick up any of these names (hdimage-base.tgz, hdimage-full.tgz, hdimage-custom.tgz) so just copy the one you want to the card with the other files listed above. In general, user contributed custom images should be called hdimage-custom.tgz

You need to do the following to get into the Maintenance Menu which is where you can update/install pdaXii13 from:

- Unplug everything from your Zaurus, ie power, cards, etc...
- Press the reset button (unlock the battery cover and press the little reset button)
- Put battery lid back on and lock the battery compartment
- Press and hold the OK key while pressing the On/Off button at the front
- You are now in the Maintenance Menu

To upgrade or flash your distro to pdaXii13, do the following from the Maintenance Menu to get into the pdaXii13 updater menu:

- Insert CF or SD (which ever has the install files)
- Plug in the power
- Choose Option 4 (Update) from the Maintenance Menu
- Select your installation source, choose CF or SD
- Confirm your selection (choose option on the left)

In the pdaXii13 updater menu choose Option 2 to repartition your Zaurus if required (depends on your current partition layout). You will need to reboot your Zaurus once you have repartitioned it and then repeat the above steps to get into the Maintenance Menu again and then into the pdaXii13 updater

menu.

From the pdaXii13 updater menu, choose Option 3 "Install kernel and bootloader" to flash your C3000 with a new kernel and boot loader image. If you have a C3100 or C3200, this option will update your existing system (which has to be pdaXrom beta3) to boot into the MicroDrive but will not update the kernel nor the rest of the system.

Once the installation has finished, you will need to extract the hdimage.tgz (or one of its variants) onto the MicroDrive. Choose Option 4 for that and wait a while until it formats your first partition as ext3 and extracts the content of the tgz file onto the first partition of your MicroDrive.

Then choose Option 6 to reboot and pdaXii13 should be installed and boot into X windows. OpenBox will be the default X window manager just like in the default pdaXrom beta3. During your first boot, your SSL private and public keys will be generated which will take a while.

If you have used the recommended partition scheme or used your own custom partition layout, then you need to format those remaining partitions. During the initial boot, you will also be given a choice to format your second and third partition if you have them. The second partition will be formatted as swap while the third partition will be formatted as FAT. Alternatively, you can do it manually as well. Start a terminal console and format the remaining partitions depending on what type of partitions you have created. For example:

```
# mkswap /dev/hda2
# swapon /dev/hda2
# mkfs.vfat /dev/hda3
# mount /dev/hda3 /data
```

The above assumes you have no CF card inserted. Otherwise use hdc instead of hda.

Once your second partition is initialised, it will be automatically enabled as swap when you reboot. Similarly, your third partition will be automatically mounted as /data after a reboot.

Installation files for pdaXii13 Spitz (Alice) can be downloaded from pdaXii13 files area on tyrannozaurus which contains a stable release version. Alternatively, there are also alpha/beta test binaries available for downloaded from the bleeding edge testing area.

pdaXii13 Spitz was build for the SL-C3000 and has only been tested on a SL-C3000. It may work on SL-C3100/SL-C3200 if they have pdaXrom beta3 or pdaXii13 Akita base installed prior to installing pdaXii13 Spitz, but may also break those models. It is not guaranteed to work, and even if you managed to get it installed, there still may be some things that won't work. **pdaXii13 Spitz** is for the SL-C3000 and runs entirely from the internal MicroDrive. It is available as pdaXii13 base and pdaXii13 full.

pdaXii13 Spitz is also the main release of pdaXii13.

pdaXii13 Akita is for SL-C1000, SL-C3100 and SL-C3200 and runs off the 128MB internal flash (NAND). It is primarily available as pdaXii13 base only, but can be upgraded to pdaXii13 full on SL-C3100 and SL-C3200, but not on the SL-C1000 due to it not having an internal MicroDrive.

Installation instructions and installation files for pdaXii13 Akita (Sally) can be found at tyrannozaurus in the akita area. The pdaXii13 Akita install process is similar to the pdaXrom beta3 install and the same that applies to pdaXrom beta1/beta3 also applies to pdaXii13 Akita installation.

It is also possible to apply all the pdaXii13 fixes to the C1000/C3100/C3200 without reflashing to pdaXii13, ie you can upgrade your existing pdaXrom beta1/beta3 setup to pdaXii13 without requiring to wipe your existing setup.

You can get the upgrade files pdaXii13-custom.tgz from tyrannozaurus. There is an install script contained in the tarball which will copy all the patched binaries and scripts to the appropriate locations when you run the script. To upgrade to pdaXii13, extract the tarball (preferrably to a CF card rather than a SD card) and run the install script (install-fix-beta3.sh) from the console outside of X.

Make sure you make a NAND backup of your Zaurus before you apply the update in case something goes wrong, then copy pdaXii13-custom.tgz to a CF card.

```
# mkdir -p /mnt/cf/upgrade
```

```
# zcat pdaXii13-custom.tgz | tar xvf - -C /mnt/cf/upgrade
# cd /mnt/cf/upgrade/custom
# /etc/rc.d/init.d/sd stop
# ./install-fix-beta3.sh / upgrade
```

This will give you something similar to pdaXii13 base.

Pre-Installed Packages

Below is a list of pre-installed applications that are part of the **pdaXii13 full** installation. The following packages are in addition to the applications installed by the default pdaXrom beta3. These applications are not all pre-installed on **pdaXii13 base**.

engines and runtimes:

- atd (scheduler)
- samba (file sharing)
- aspell (spelling engine)
- scim (chinese and japanese support)
- x11vnc (vnc server)
- python (interpretor)
- perl (interpretor)
- java (classpath and jamvm)
- gsnapshot (screen capture)
- xscreensaver (screen saver)
- wallpaper (random background images)
- gnome-bluetooth (obex gui)
- gnome-phone-manager (gnokii frontend)
- festival/mbrola (text to speech)

useful command line utilities:

- vim (replacement for vi)
- nano (text editor)
- scrot (screen capture)
- lame (mp3 encoder)
- feh (image viewer)
- mplayer (video player)
- gplflash (flash player)
- flite (speech synthesizer)
- bzip2 (compression utility)
- unrar (uncompression utility)
- unzip (uncompression utility)
- diffutils (file comparison utility)
- ipktools (ipk building tools)
- xdialog (dialog tools)

applications:

- abiword (word processor)
- gnumeric (spreadsheet)
- xournal (note taking)
- medit (text editor)
- textedit (text editor)
- sylpheed (email)
- firefox (browser)
- dillo (browser)
- links (browser)
- gftp (ftp client)
- amule (p2p)
- gaim (IM client)
- wifi-radar (wifi detector)
- kismet (network sniffer)

- tightvnc (vnc client)
- putty (net client)
- mrxvt (tabbed terminal emulator)
- multi-aterm (tabbed terminal emulator)
- rox (file manager)
- xfe (file manager)
- pcmanfm (file manager)
- emelfm2 (file manager)
- LinNeighborhood (samba client frontend)
- xmmsplayer (video player plugin for xmms)
- gmplayer (mplayer with GUI)
- smplayer (GUI for mplayer)
- vlc (streaming media)
- streamtuner/streamripper
- xarchiver (file compressor/decompressor GUI)
- gqview (image viewer)
- gthumb (image viewer)
- scribble (simple drawing)
- qbedic (dictionary)
- stardict (dictionary)
- epdfview (pdf viewer)
- evince (pdf viewer)
- fbreader (ebook viewer)
- justreader (ebook viewer)
- opie-reader (ebook viewer)
- qcomicbook (comic viewer)
- kchmviewer (chm viewer)
- dia (flow diagrams)
- gimp (graphics)
- khdrecord (sound recording)
- gpe-calendar (calendar)
- gpe-announce (alarms)
- gpe-contacts (addressbook)

games:

- xdemineur (mines)
- tetrix (tetris)
- tictac (tictactoe)
- xpat2 (solitaire)
- xinvaders (space invaders)
- xkobo

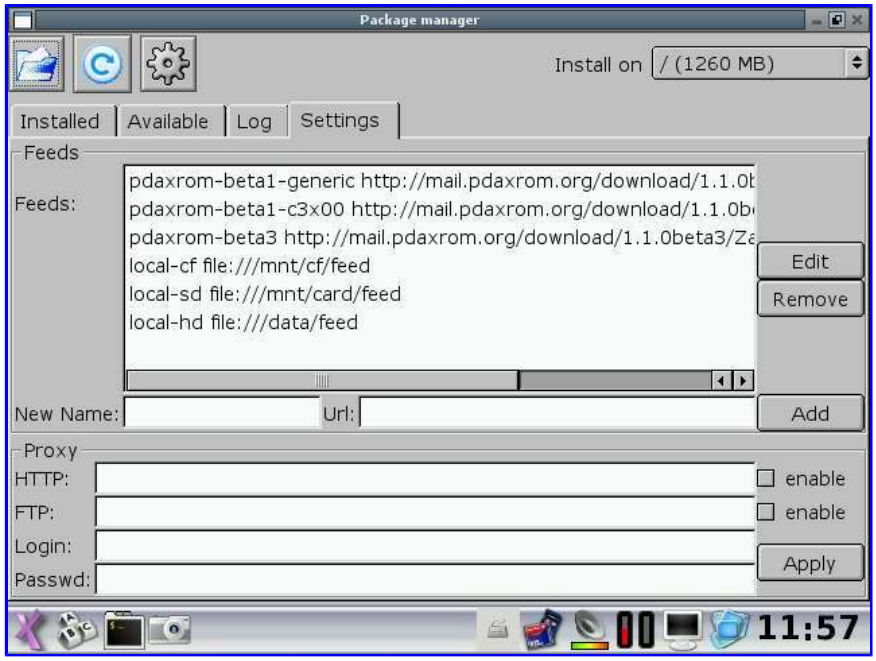
There are many more packages beside the ones on the official feeds. Many user contributed packages exist and can also be installed. beta1/beta3 packages are compatible with pdaXii13. Some packages for pdaXrom beta4 and r121 will also work on pdaXii13.

Configuring Feeds and Installing Packages

The Package Manager has been pre-configured to point to a special beta3 feed (http://www.tyrannozaurus.com/feed/beta3/feed/) which is a merger of several beta3 compatible feeds including the official beta3 and beta1 feeds as well as some custom feeds. It also points to local feeds on /mnt/card/feed, /mnt/cf/feed and /data/feed.

To create your own custom local feed, you can download the beta1 feed.tar.gz files for the generic and C3x00 specific one.

You can then merge the two feeds (and any additional ones) onto your internal MicroDrive or one of the external cards. Just extract the files into a directory (extract the 7x0 archive first and then the C3100 one on top of it to eliminate the duplicates). Then use **ipkg-make-index** to generate a new *Packages* file for the feed.



You can also install individual packages without requiring a feed, but you will need to manually install dependant packages yourself. You can either use the GUI Package Manager or use the command line **ipkg** command.

Keyboard Shortcuts

The pdaXii13 custom keyboard layout (spitz.xmodmap) is fully functional with all keys mapped. Below is a list of keyboard shortcuts and customisations that exists on pdaXii13.

There are two types of changes to the keymappings. The first one are the essential fixes to make the keymap work, and the second ones are enhancements to make it more usable and similar to the keyhelper enhancements on the Sharp distro.

To customise the default keymap, modify /etc/sysconfig/keyboard/kernel.map and /etc/X11/kb/spitz.xmodmap and use *loadkeys* and *xmodmap* to reload the key mappings respectively. Use *xev* within X to determine the keycode for the keys. If you don't like my custom keymap, then use the original akita.xmodmap instead.

Sticky keys are automatically enabled and can be controlled with the **ax** command which controls all the AccessX functionality. With sticky keys, you can hit the control/modifier keys, ie: Fn, Shift, Ctrl and Alt, once and then hit the second key without needing to keep the control/modifier key pressed. For example, to produce A, simply press *Shift* followed by *a*. This is very nifty for one handed typing.

Most keys are mapped as displayed on the keyboard except for some additional keys which have also been mapped but are not marked on the keyboard. Also some additional keys have been remapped as follows:

- Fn+m = Scandinavian A with loop
- Fn+o = Danish O crossed out
- Fn+p = British Pound sign
- Fn+a = AE character
- Fn+s = Euro symbol
- Fn+q = Backtick (accent grave)
- LeftKanji = Alt (Alt_L)
- RightKanji = Super (Super_L)

By defeault, there are also several predefined key combos which do the following:

- Ctrl+Alt+BS = Shutdown X
- Ctrl+Alt+Left = Switch to previous Virtual Desktop
- Ctrl+Alt+Right = Switch to next Virtual Desktop
- Shift+Alt+Left = Send to previous Desktop
- Shift+Alt+Right = Send to next Desktop

- Ctrl+Alt+d = Toggle Show Desktop
- Ctrl+Alt+m = Toggle Maximize/Restore Window
- Fn+Ctrl+Alt+Down = MoveRelativeVert
- Fn+Ctrl+Alt+Up = MoveRelativeVert
- Fn+Ctrl+Alt+Left = MoveRelativeHorz
- Fn+Ctrl+Alt+Right = MoveRelativeHorz
- Alt+Tab = Switch forward between running Applications
- Alt+Shift+Tab = Switch backward between running Applications
- Alt+0 = MaximizeFull
- Alt+5 = UnmaximizeFull
- Alt+9 = Iconify
- Alt+8 = Resize
- Alt+7 = Move
- Alt+4 = Close
- Alt+6 = ToggleShade
- Menu = Activate Launch Menu
- Home = Mode
- Mail = rox
- Address = xmms
- Calendar = mrxvt
- Shift+Calendar = aterm
- Ctrl+Alt+c = screencap
- Shift+Ctrl+c = screencap

Some function keys have been remapped as follows:

- Super+1 = Switch VGA mode (320x240)
- Super+2 = Switch SVGA mode (640x480)
- Super+3 = Decrease Brightness
- Super+4 = Increase Brightness
- Super+5 = Refresh

The silkscreen keys are mapped as follows by default:

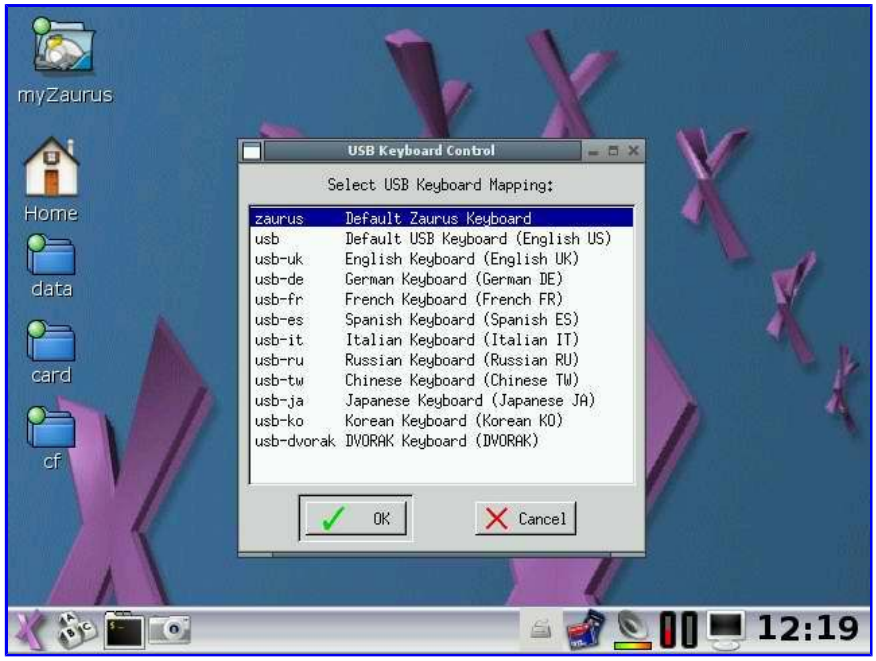
- Home = dilllo
- Mail = sylpheed
- Address = xmms
- Calendar = rox
- Dictionary = stardict

To reduce finger movements, I also added extra mappings so that instead of pressing Fn+Ctrl+Alt, you can also use the Home key instead, and instead of Ctrl+Alt, you can use the Super key (right kanji key). I also associated most Alt+key combos to be also activated by Home+key as well.

The above changes can be done by modifying *rc.xml* which is by default located under */etc/xdg/openbox/*, but may also be located under /home/root/.config/openbox/. In pdaXii13, those two locations are symbolically linked.The silkscreen and shortcut keys at the bottom of the keyboard can also be changed using the **Input Setup** tool.

SCIM input language switching can be activated by pressing *Ctrl+space* or *Ctrl+/*

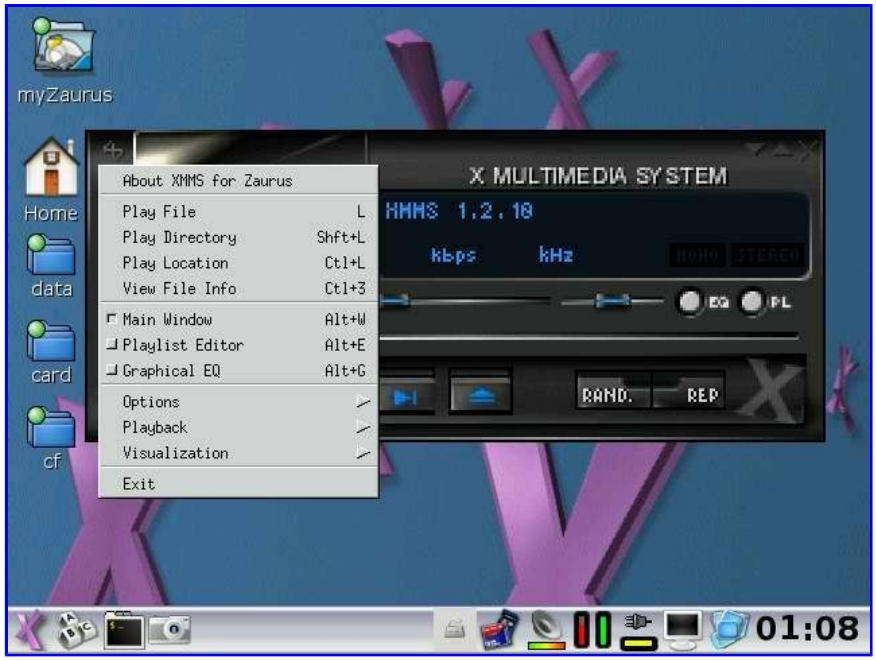
When you plug in a USB keyboard, you can automatically remap the keys so that the keys on the external keyboard are mapped correctly, but it will result in the Zaurus keyboard being mismapped. When the external keyboard is unplugged, the keymapping for the zaurus keyboard is restored. This is handled by the hotplug subsystem.



You can also manually switch the keyboard mapping for your USB keyboard by using the **USB Keyboard Selector** tool. It allows you to switch between the default Zaurus keymapping (spitz.xmodmap) and the default USB keyboard mapping (usb.xmodmap). There are additional options for other keyboard mappings which you can add by placing keyboard maps for those mappings under /etc/X11/kb/ and naming your keyboard mapping file usb-LANGUAGE.xmodmap where LANGUAGE is the two letter abbreviation in lowercase.

The default XMMS keys have been changed to make it more Zaurus friendly, especially when the Zaurus is in portrait mode and you want to use the buttons and the scrollwheel at the back of the Zaurus to control XMMS just like the MusicPlayer in Sharp ROM. The following keys can be used to control XMMS:

- X or OK = Play
- C or Cancel = Pause
- V = Stop
- N = Increase Volume
- M = Decrease Volume
- Left = Skip Backward
- Right = Skip Forward
- Z or Up = Prev Song
- B or Down = Next Song
- R = Repeat
- S = Shuffle
- L = Load track to playlist
- Shift + L = Load folder
- Alt + S = Skin browser
- Ctrl + P = Preferences



xpdf has the following default key mappings:

- o = open file
- r = reload
- f = find next
- n = next page
- p + previous page
- + = zoom in
- - = zoom out
- z = zoom page
- w = zoom page width
- 0 = zoom normal
- q = quit

Below are some of the keyboard shortcuts for mrxvt:

- Ctrl+Shift+t = Create a new tab
- Ctrl+Shift+n = Create a new tab
- Ctrl+Shift+w = Close active tab
- Ctrl+Shift+q = Close all tabs and exit
- Ctrl+Fn+Up = Activate left tab
- Shift+Left = Activate left tab
- Ctrl+Shift+h = Activate left tab
- Ctrl+Shift+p = Activate previous active tab
- Ctrl+Tab = Activate previous active tab
- Ctrl+Fn+Down = Activate right tab
- Shift+Right = Activate right tab
- Ctrl+Shift+l = Activate right tab
- Ctrl+Shift+Left = Move active tab to left
- Ctrl+Shift+less_than = Move active tab left
- Ctrl+Shift+Right = Move active tab to right
- Ctrl+Shift+greater_than = Move active tab right
- Ctrl+equal = Increase font size
- Ctrl+minus = Decrease font size
- Ctrl+Shift+plus = Increase font size by 2
- Ctrl+Shift+underscore = Decrease font size by 2
- Ctrl+Shift+r = Toggle pseudo-transparency
- Ctrl+Shift+i = Hide/show tabbar
- Ctrl+Shift+s = Hide/show scrollbar
- Ctrl+Shift+m = Hide/show menubar
- Ctrl+Shift+a = Hide/show tabbar buttons
- Ctrl+Shift+f = Toggle full screen mode
- Shift+Insert = Paste X selection into active tab.
- Ctrl+Shift+v = Paste X selection into active tab.

- Shift+Up = Scroll up one line
- Shift+Down = Scroll down one line

Some of the sticky keys will also work outside of X as well. sticky keys are enabled by default. You can disable them by settings STICKY_KEYS to false in pdaxii13.conf and restarting X.

Screen Rotation

Screen rotation works automatically on pdaXii13. Openbox will detect transformation between laptop mode (landscape) and PDA mode (portrait) and rotate the screen by calling **rotate.sh** which will ensure it is done appropriately by checking the current orientation and calling the relevant commands accordingly.

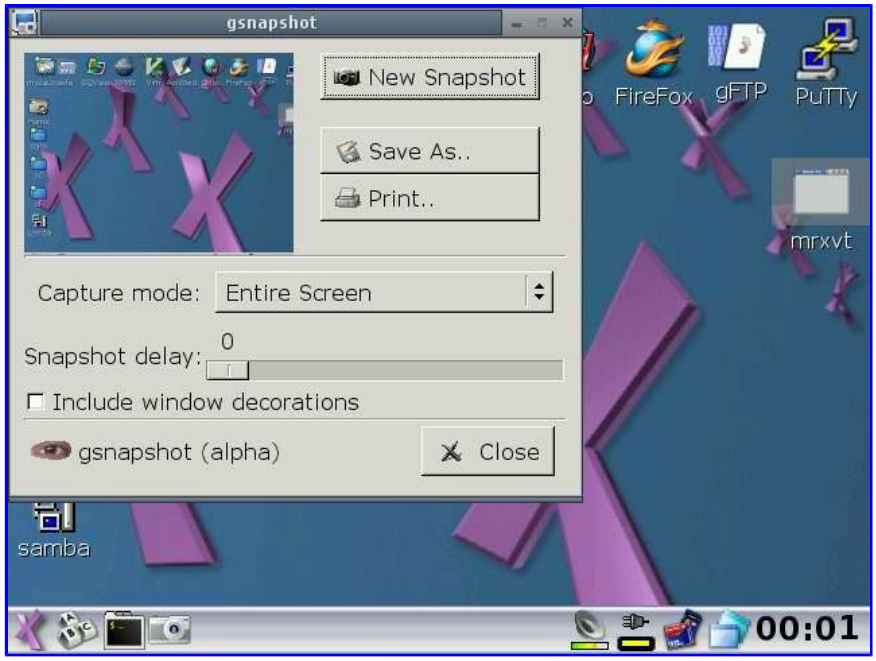
You will end up with the taskbar at the correct location for the orientation you are in. The currently running applications will also be resized to fit your orientation. In order for this to happen, the running applications will be maximized. The rotate option on the menu will also call rotate.sh and force a manual rotation.

Screen Capture

A screenshot can be taken by running xwd -display :0 > screendump.xwd

A smarter script (screencap) has been written that mimics the capture behaviour on the Sharp distro, ie. the script is bound to the Ctrl+Alt+c key combo and activated when that key sequence is pressed. A shutter sound (click.wav) is then played before doing the screen dump. The filenaming and sequencing has been made to mimic the Sharp snapshot tool. See the **screencap** script.

If scrot is installed it will use scrot to capture a png image, otherwise xwd is used instead and if ImageMagick is installed, then it will be used to convert the xwd image into a png image.



In pdaXii13 full, an additional screencap icon is placed on the taskbar and once tapped will wait 10 seconds before taking a screenshot. Alternatively, you can also use **gsnapshot** which allows you to capture a specific area of the screen.

Desktop and Menu Customisation

The .desktop files defining the applications are located under */usr/share/applications* and the corresponding icons are located under */usr/share/pixmaps*

The desktop is by default managed by matchbox desktop in pdaXrom. pdaXii13 full defaults to rox pinboard instead. You can either access the application icons via the menu or the desktop panels/pinboard.

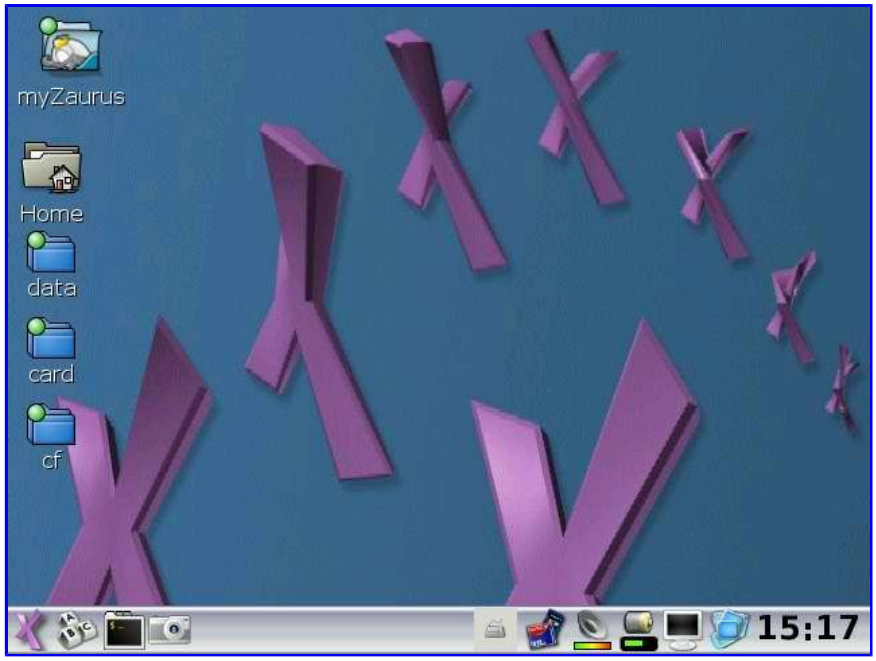
You can manually replace the matchbox desktop with rox filer to get an integrated file manager and desktop manager in pdaXrom. To do that, modify */home/root/.xinitrc* and replace the following:

```
eval "matchbox-desktop $MBBGND" 2>/dev/null >/dev/null &
```

with this:

```
eval "rox --pinboard=MYPINBOARD" &
```

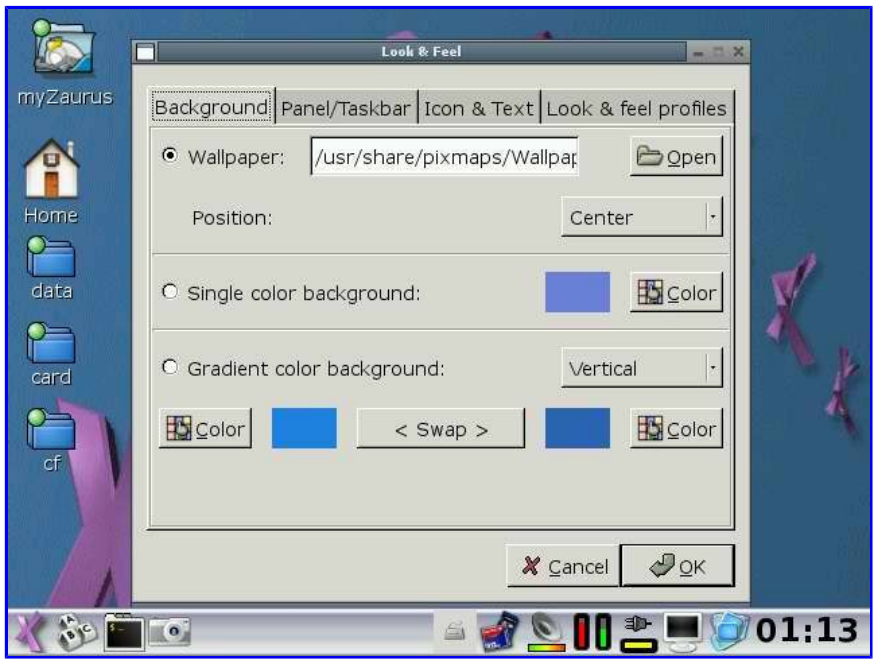
and then restart X.



The config files for ROX are under */usr/apps* and */home/root/Choices/ROXFiler*

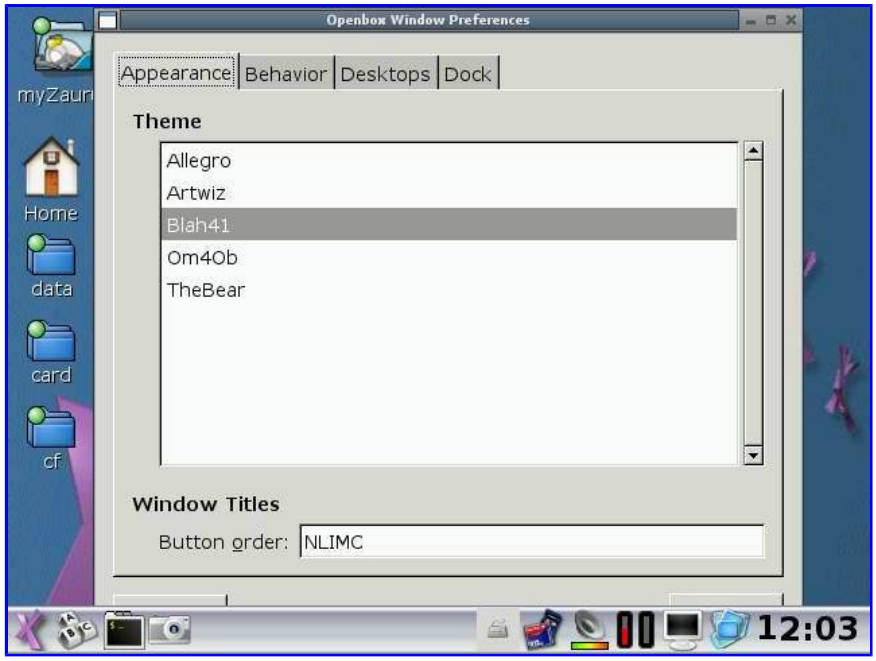
In pdaXii13, the above change has been implemented in such a way that rox is used if it is installed, but the default matchbox desktop is used instead if rox is not installed. You can also specify whether to use matchbox desktop or rox pinboard in the pdaXii13 config file called pdaxii13.conf located under /home/root/Choices

The Look & Feel tool in the Systems Tools can be used to change the default wallpaper as well as change many other aspects of how the GUI looks like. The wallpaper can be any png image file, but a resolution of 640x440 is ideal.



However, if you changed your desktop to use rox filer instead of the matchbox desktop, then you can right click (Fn+tap) to activate the rox filer properties where you can drag and drop an image into the background section. There is also a command line tool called **rox-bset** which you can use to replace the background as well as enabling random backgrounds.

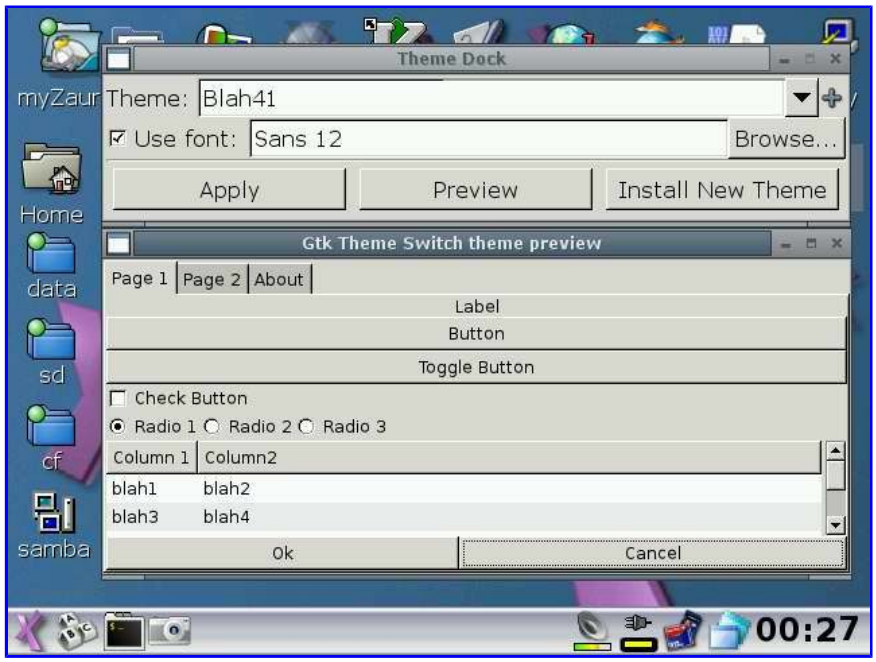
To change the theme, use the **OpenBox Config** tool (obconf) which can configure other settings in OpenBox as well.



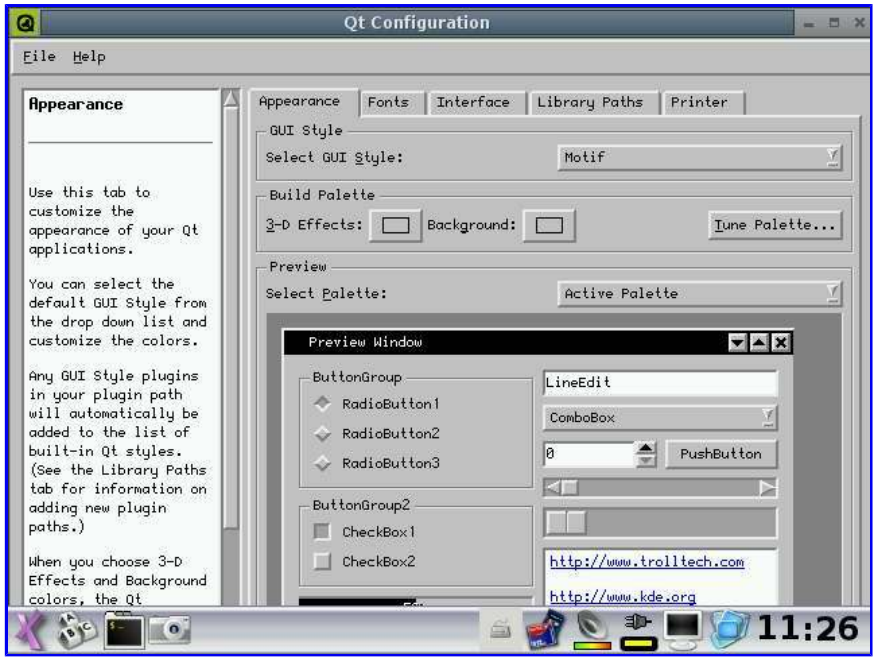
To apply manual changes done by editing rc.xml, either restart X or run the following command:

```
# openbox --replace 2>/dev/null &
```

There is also an additional **GTK Theme selector** in pdaXii13 to allow you to change the GTK2 theme used by some applications. The GTK2 theme also controls the font size for those applications. The GTK Theme selector also allows you to just change the font without switching themes.

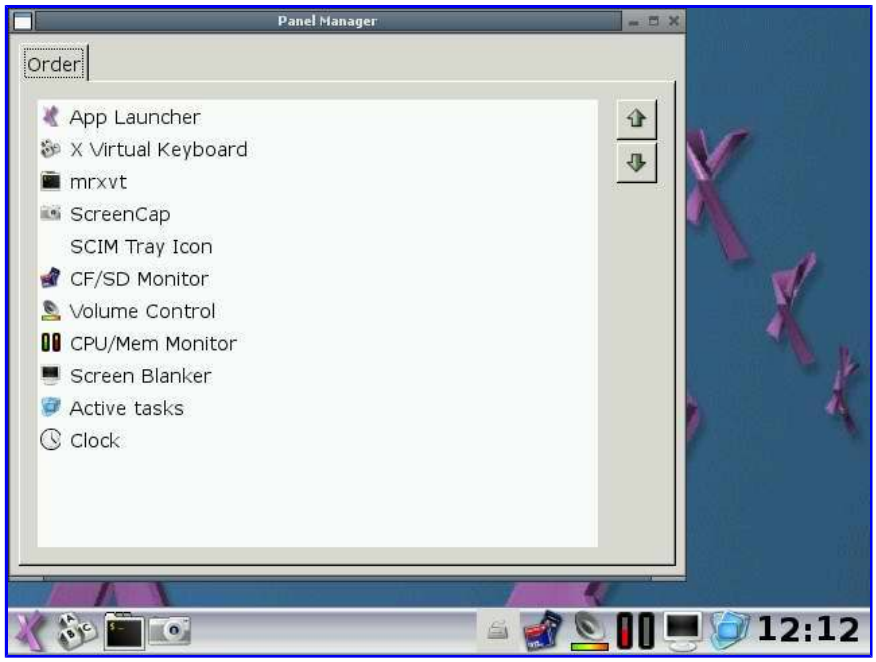


The **QtConfig** tool can be used to change the display settings for Qt applications.



Most applications in pdaXrom are GTK based, but there are also a few QT applications.

Finally, in pdaXii13, the order of the icons on the toolbar can be changed with the **Panel Manager** tool.

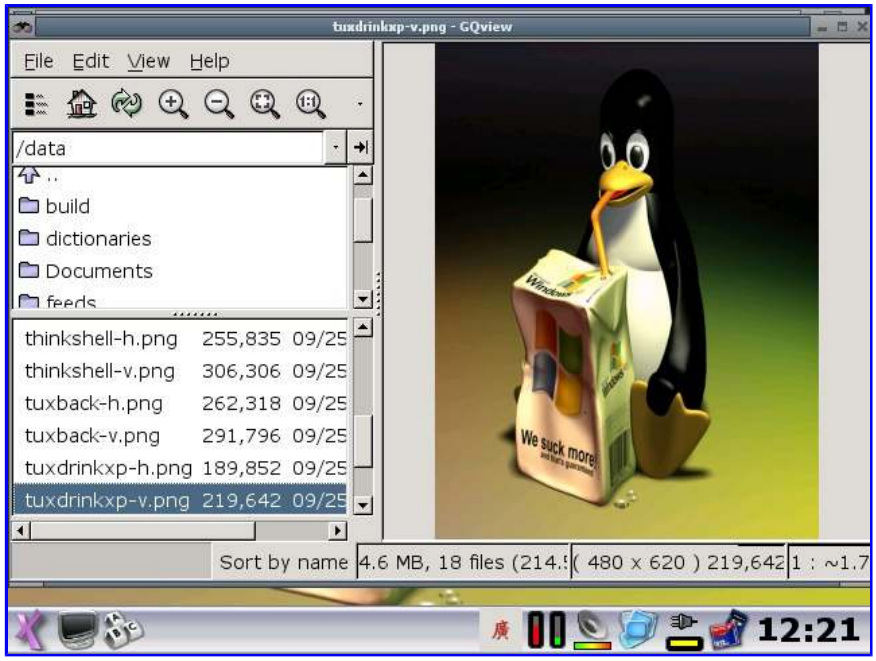


This tools is new to pdaXii13 and not in the default pdaXrom but can be added manually.

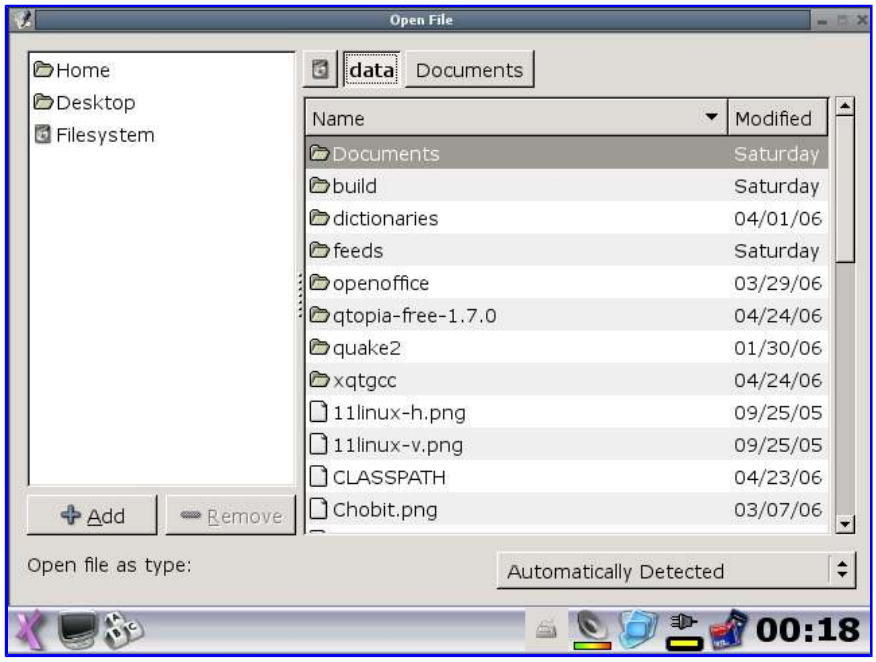
Window and Dialog Sizes

There are quite a few applications which open up in a window that is way too big for the Zaurus. This can usually be worked around by maximising the window in question (press *Home+m* or *Super+m*).

Some application can be tailored to a specific window size by editing their config files. Other applications have the size hardcoded. I have recompiled a few such applications, ie GQView and AbiWord.



The File Open Dialog is also quite big and this problem is across most GTK2 applications. I have recompiled GTK2 and hacked the FileChooser to be more suitable for the Zaurus screen. You can either install my updated GTK2 package or replace libgtk-2.0.so.0.600.2 manually.



pdaXii13 has this fix applied by default and most windows will fit onto the display rather than extend beyond the physical screen size.

Window Managers

pdaXrom uses openbox as its default window manager because it is lightweight and loads pretty fast. However, it does not automatically resize applications that are oversized. Suitable alternatives seem to be icewm and fluxbox which are also included in pdaXii13 full.

openbox (default with matchbox desktop)

openbox (with rox pinboard)

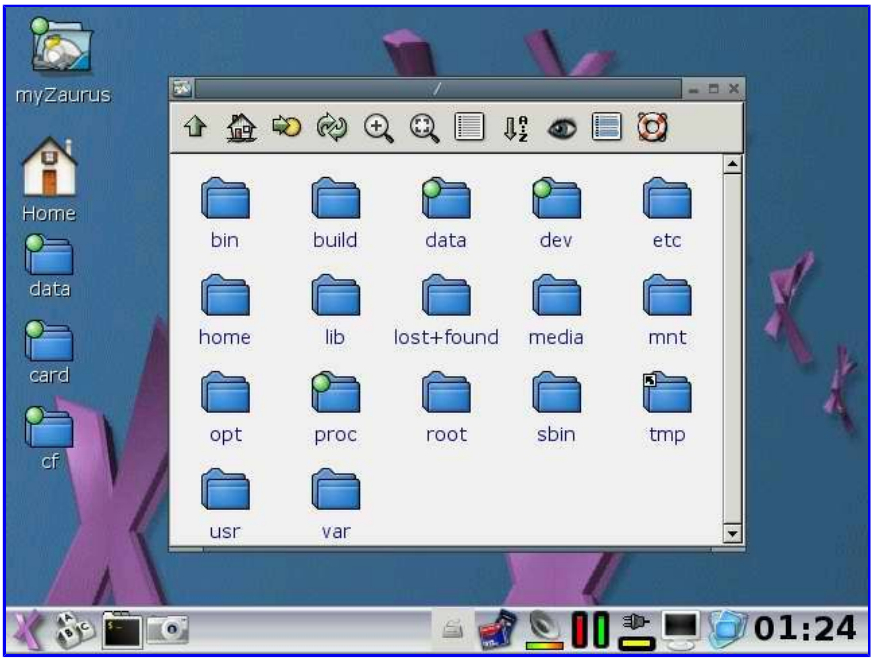
icewm (with rox pinboard)

fluxbox (with rox pinboard)

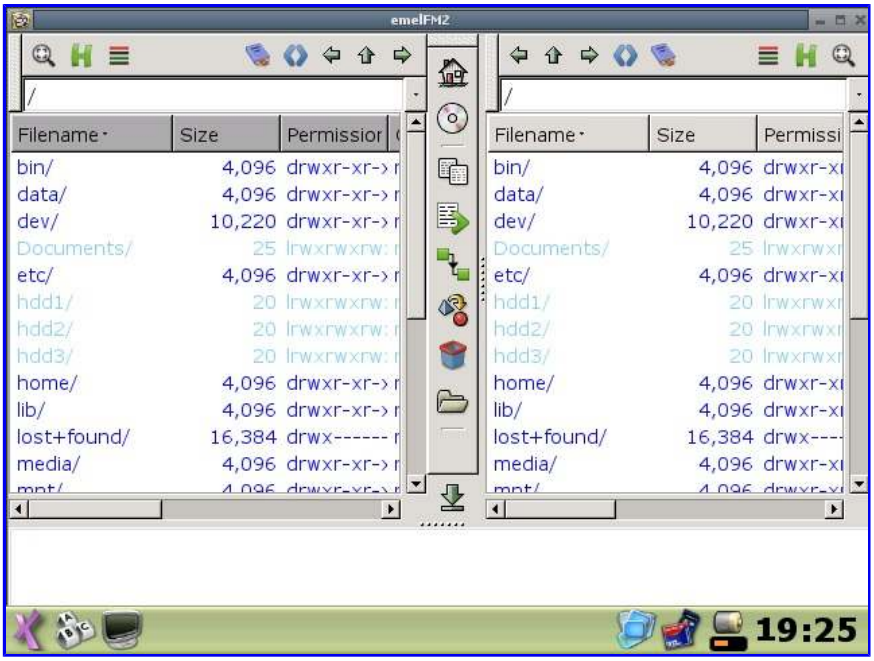
The default window manager in pdaXii13 is openbox, just like in pdaXrom. However, once you exit X, you can use **xselect** to switch and make fluxbox or icewm your default window manager which will run when you start X again.

File Managers

There is no default GUI file manager in the default pdaXrom, however, rox filer is a good candidate for this function and has been included in pdaXii13.



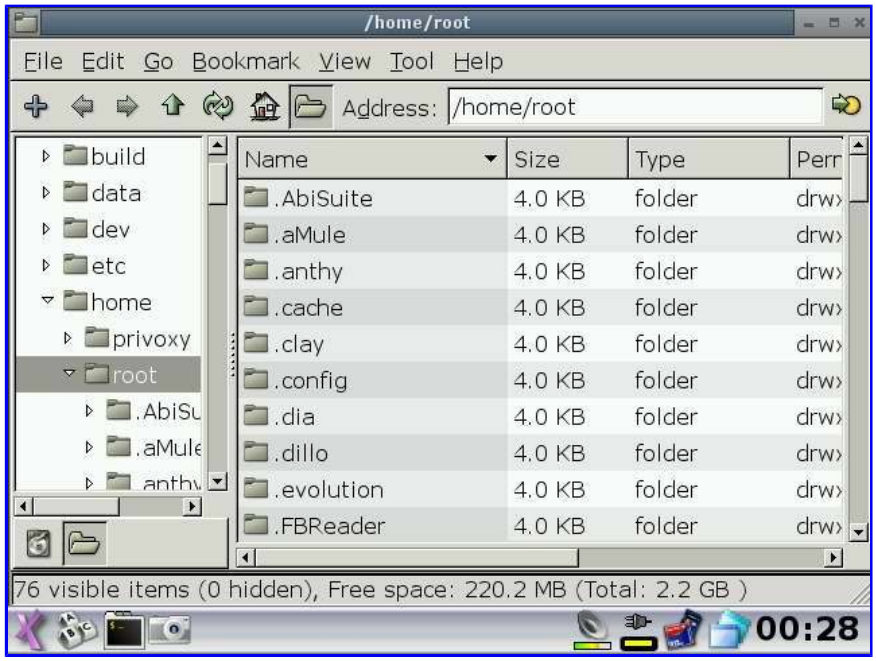
In addition, emelFM2 is also a good file manager with two panels and has also been included in pdaXii13.



Midnight Commander (mc) is a console based file manager that is by default installed in pdaXrom and has been updated in pdaXii13.

mrxvt
Terminal
Left File Command Options Right
Hint: Need to quote a character? Use Control-q and the character.
root-:)
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn
00:33

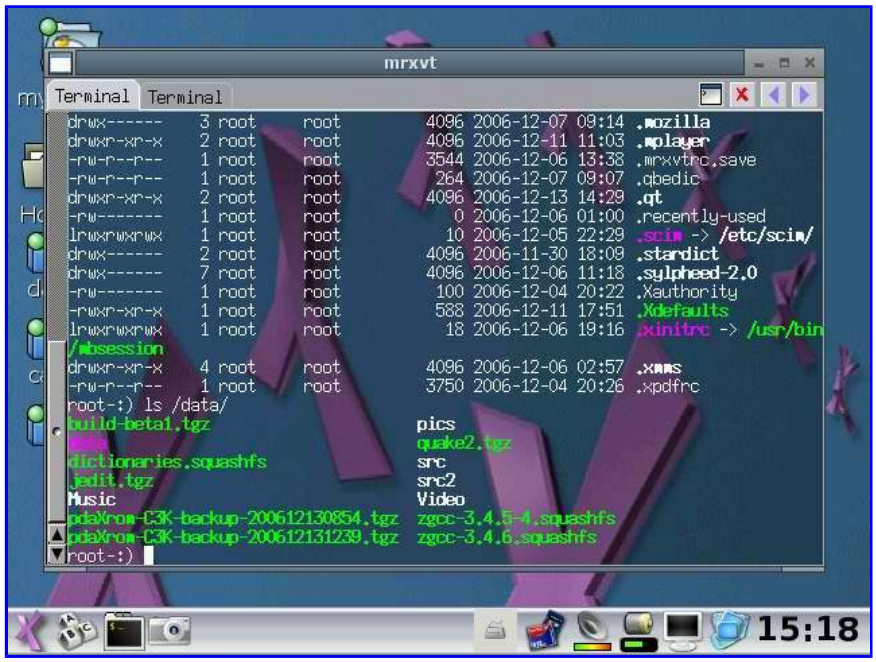
Additionally, pcmanfm which is a fast and efficient file manager is also included.



And lastly, xfe which is much like windows explorer has also been included in pdaXii13.

Terminals

mrxvt is the default terminal in pdaXii13 because it looks nice and has pseudo transparent background support. To enable transparency, copy Xdefaults to /home/root as .Xdefaults if mrxvt does not have transparency enabled by default. The Xdefault file enables transparency for mrxvt, rxvt, aterm and multi-aterm.


mrxvt
Terminal Terminal
drwx------ 3 root root 4096 2006-12-07 09:14 .mozilla
drwxr-xr-x 2 root root 4096 2006-12-11 11:03 .mplayer
-rw-r--r-- 1 root root 3544 2006-12-06 13:38 .mrxvtrc.save
-rw-r--r-- 1 root root 264 2006-12-07 09:07 .qbedic
drwxr-xr-x 2 root root 4096 2006-12-13 14:29 .qt
-rw------- 1 root root 0 2006-12-06 01:00 .recently-used
lrwxrwxrwx 1 root root 10 2006-12-05 22:29 .scim -> /etc/scim/
drwx------ 2 root root 4096 2006-11-30 18:09 .stardict
drwx------ 7 root root 4096 2006-12-06 11:18 .sylpheed-2.0
-rw------- 1 root root 100 2006-12-04 20:22 .Xauthority
-rwxr-xr-x 1 root root 588 2006-12-11 17:51 .Xdefaults
lrwxrwxrwx 1 root root 18 2006-12-06 19:16 .xinitrc -> /usr/bin
/xbsession
drwxr-xr-x 4 root root 4096 2006-12-06 02:57 .xmms
-rw-r--r-- 1 root root 3750 2006-12-04 20:26 .xpdfrc
root-:) ls /data/
build-beta1.tgz
dictionaries.squashfs
jedit.tgz
Music
pdaXrom-C3K-backup-200612130854.tgz
pdaXrom-C3K-backup-200612131239.tgz
pics
quake2.tgz
src
src2
Video
zgcc-3.4.5-4.squashfs
zgcc-3.4.6.squashfs
root-:)
15:18

aterm and multi-aterm, as well as rxvt are also available as alternative terminals and installed on pdaXii13 by default. xterm which is a bit broken in pdaXrom has been fixed in pdaXii13 and displays the fonts without the extra boxes.

Zaurus Backup

You should always backup your system since that is the only way to recover if something goes wrong. The easiest way to backup is to archive everything using **tar** and then use **gzip** to compress the archive.

I have created a script called **zbackup** which backs up the system and its configurations so you can use the backup image generated by it to restore your system. The generated tgz file can be used to restore pdaXii13 by simply renaming it to hdimage-custom.tgz and flashing it.

It is recommended to exit X before doing a backup. By default, the backup file will be placed under /data

/opt, /data and /mnt will not be backed up as well as any other additional directories that you create on /. zbackup has been designed to only work on the SL-C3000.

zaurus user

pdaXii13 has a zaurus user to run certain services such as samba, xscreensaver and amule where security is important, but for most other things, the root user is used. This is a slight security concern, but better than pdaXrom where everything is run by root. Future releases might fix this up to make everything run by the zaurus user instead just like in Sharp/Cacko.

Resume and Suspend

suspend works when invoked from the start menu or when the On/Off button on the front side is pressed. resume also works just like in Sharp ROM where you need to press the On/Off button when the Z is suspended. It also appears that the Z does not like to resume unless you had closed the lid and reopened it. You can cheat by pressing the little nodge at the base of the swivel with your stylus. You may need to wait ten seconds before the Zaurus resumes.

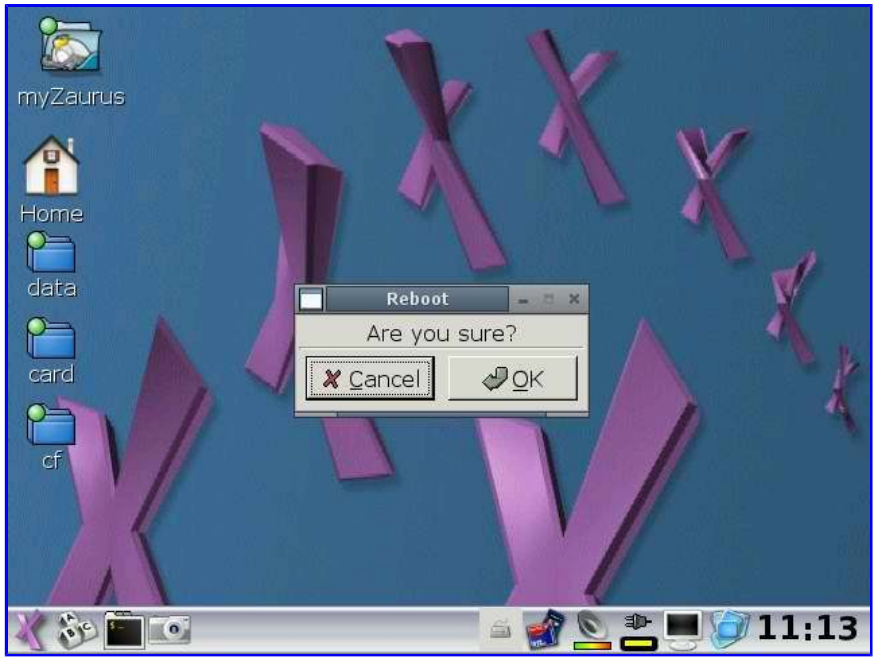
The hdd led is no longer left on when suspending except when the MicroDrive's filesystem is really badly corrupted and in need of a manual fsck which happens occasionally if you have not rebooted the Zaurus for a long time.

In pdaXii13, the sound daemon (esd) has been configured not to respawn which prevents it from locking up the keys on resume. It is also advisable to add **esdctl off** into the apm suspend script and have **esdctl on** in the corresponding resume script to automatically suspend and resume the sound daemon. This is already done in pdaXii13 and allows the Zaurus to be suspended while XMMS is actively playing without locking out the sound device. On resume, XMMS is able to continue playing. In the extreme case when the sound device does get locked, you can run the **fixdsp** command to rectify the problem or reboot the Zaurus.

The blanker applet has also been fixed and can now really enable and disable the DPMS suspend feature. It also calls the xset-wrapper script which can be used to customise the toggle for suspend/screensaver behaviour.

Reboot

pdaXii13 has a reboot option in the menu. This is added by creating a reboot.desktop under /usr/share/applications with the corresponding reboot.png under /usr/share/pixmaps as well as the **Reboot** binary under /usr/bin



When you select the reboot option from the menu, you will get prompted to confirm just like when you select exit to quit X.

fsck

fsck is run automatically each time you reboot. However, if your filesystem is too messed up, you will need to run fsck manually. In most cases, you will be thrown to a terminal shell prompt during boot if a manual fsck is required. Run the following command when this happens:

```
# e2fsck -y /dev/hda1
```

The above assumes you have not CF card inserted. Once fsck has finished running, reset your Zaurus by pressing the reset button inside the battery compartment.

It is recommended to reboot the Zaurus regularly so fsck is run to check for inode corruptions. This is required because the Zaurus implements a kind of lazy inode journalising. If you install or de-install lots of packages, it is also recommended to reboot afterwards.

pdaXii13 can be booted into Maintenance Mode, ie this is a special stage before the MicroDrive is mounted so you can run fsck on it. The pdaXii13 Config tool will allow you to do enable Maintenance Mode.

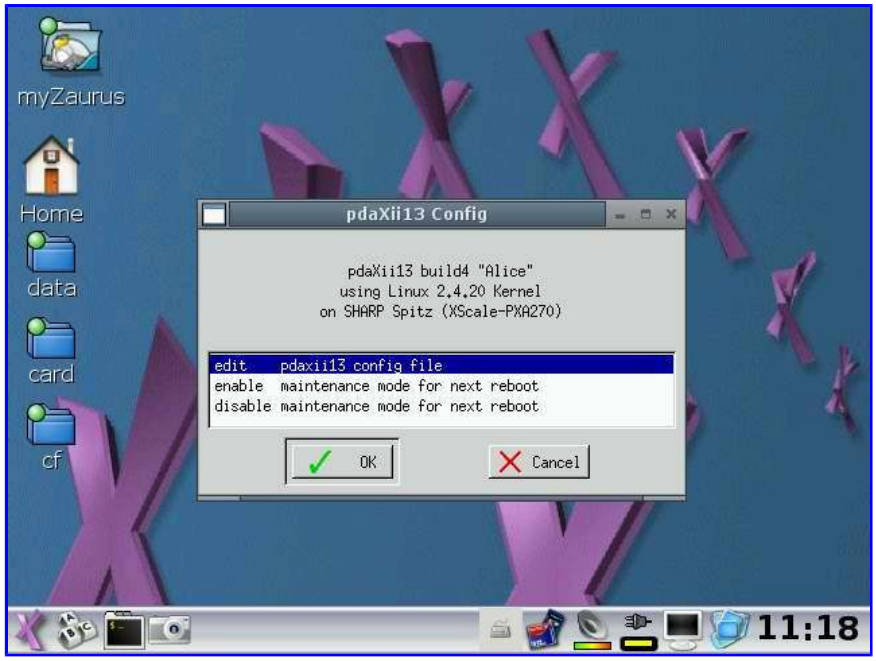
You can also manually fsck your MicroDrive by booting into the emergency partition using D+B key sequence and running fsck from there.

Alternatively, you could start the pdaXii13 installer and use option 5 to go to a console and run fsck from there.

You could also use the fsck feature in the Maintenance Menu, so after you press On/Off while pressing the OK button, select Option 2 (data check) and then select Option 2 (run fsck) again.

pdaXii13 config

The file /home/root/Choices/pdaxii13.conf defines some rudimentary startup configurations for pdaXii13 such as whether to start X when booting up, whether to play the welcome sound, whether to load SCIM automatically, whether to load ROX or the default matchbox desktop, and which mplayer to invoke from ROX filer as well as whether to load torsmo or activate sticky keys.



The special Maintenance Mode can also be enabled from this config tool. When Maintenance Mode is enabled, the next time you reboot your Zaurus, it drops into a special shell before the MicroDrive is mounted so you can manually fsck it. Once you are done, you need to reset your Zaurus by pressing the little reset button inside the battery compartment.

auto mounting

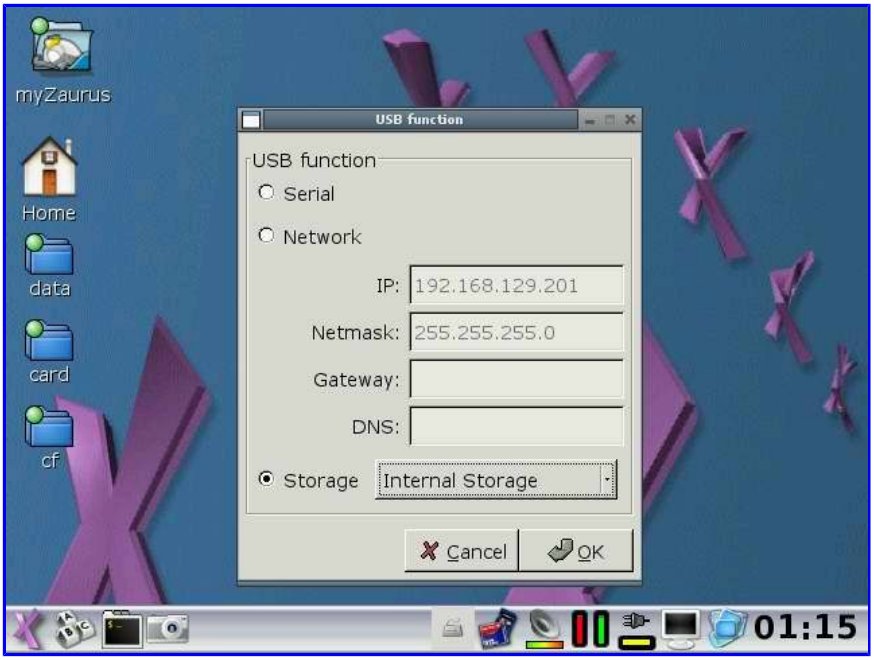
Automatic mounting via /etc/fstab works in pdaXii13. It does not work in the pdaXrom default config, but I integrated the features from my automount scripts into the default rc scripts. It creates up to 8 loop devices instead of the default 2 loop devices. Furthermore, loop images are also automatically mounted at bootup.

USB disks are also mounted automatically once they are plugged in. Up to four USB disks can be mounted automatically if you have a hub.

You can also manually mount and unmount the SD card, CF card and one USB storage device. The card applet has been enhanced to allow this.

USB network/storage

The USB module works in two modes similar to the Sharp distro. It can either allow the Zaurus to be used as a USB disk or used as a virtual network adaptor. The Zaurus can be mounted as a USB storage device when connected with the USB sync cable. It will allow you to either share the SD, CF or internal MicroDrive (partition 3 which is mounted as /data).



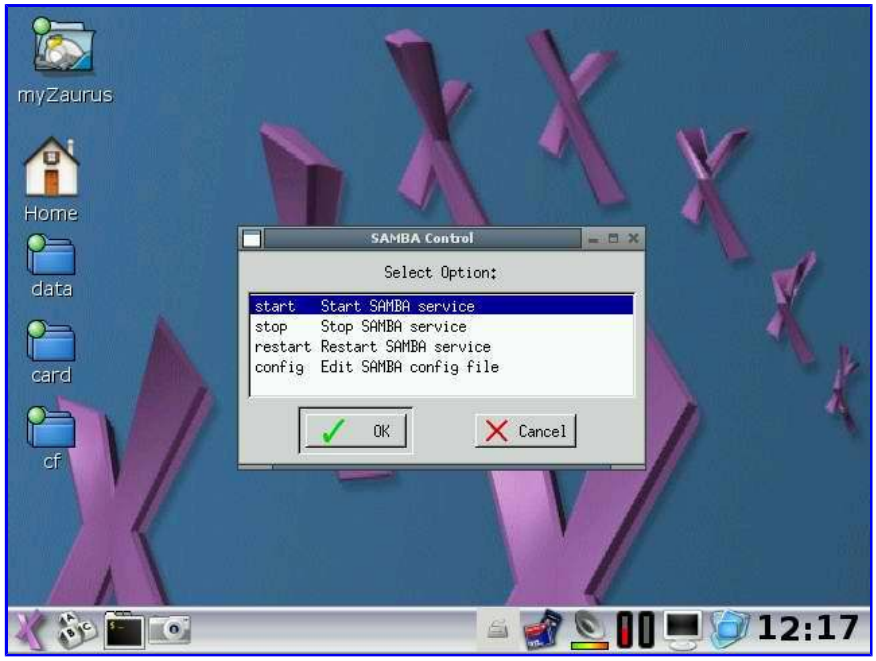
Alternatively, the USB network mode will allow you to connect the Zaurus via a virtual ethernet network and thus will allow you to share files via Samba and also allow you to use any TCP/IP dependant network services if enabled. If for example the PC or Laptop you connect to via USB cable has internet sharing (Windows ICS or iptables on Linux) configured, then the Zaurus can access the internet through that connection. See the networking section of my customisation guide for Sharp ROM on how to do that. I have also written a basic script **usbnet** which automatically configures the usbd0 interface to use the Windows ICS (Internet Connection Sharing).

Similarly, I have also added a script called **usbstorage** which quickly switches the USB mode back to storage mode using the previously configured options. Using **usbnet** and **usbstorage** you can quickly switch between USB network and USB storage mode without going into the USB config GUI.

Samba

Samba is pre-installled and configured to work on the defined interfaces as a server so you can mount the Zaurus as a samba/windows share. It may need to be reloaded if you create additional network connections in order to share files on those newly created connections as well. To restart the samba service, you can type: **/etc/rc.d/init.d/samba restart** or simply **samba restart**

I also created a simple GUI to control samba.



samba runs as the zaurus user and inherits the rights and file permissions of that user.

The Zaurus also has a samba client, so it can mount other machines that have samba enabled, even Windows machines. Additionally, the LinNeighborhood interface allow you to mount samba/windows shares from a GUI.

Bluetooth

The bluetooth sub-system included with pdaXii13 uses gnome-bluetooth as the GUI interface for file transfers via the OBEX protocol.

Files can be send to a mobile phone from rox's SendTo interface which invokes the gnome-bluetooth interface to give you a graphical tool to select the bluetooth destination device followed with a graphical progress dialog for showing the progress of the file transfer.

If the gnome-bluetooth receiving interface is enabled, then upon a request for receiving a file, a GUI is displayed to either allow or deny the file.

Additionally, a bluetooth phone manager GUI is also available which uses gnokii underneath to communicate with the mobile.

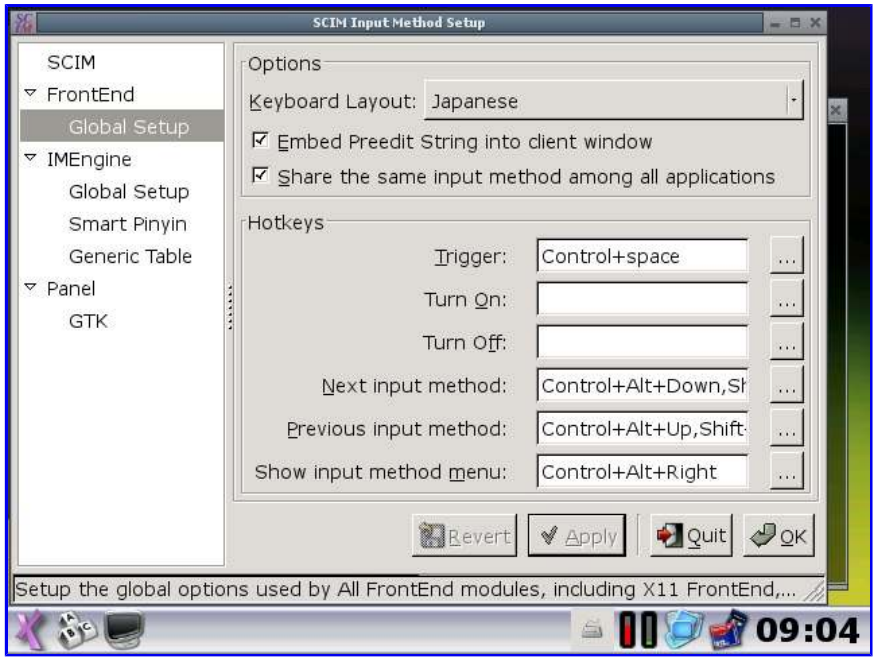
WiFi

Wireless networking is also supported in pdaXii13. Some extra firmware is included in pdaXii13 to support wifi devices that require a firmware to be uploaded to the device to initialise it.

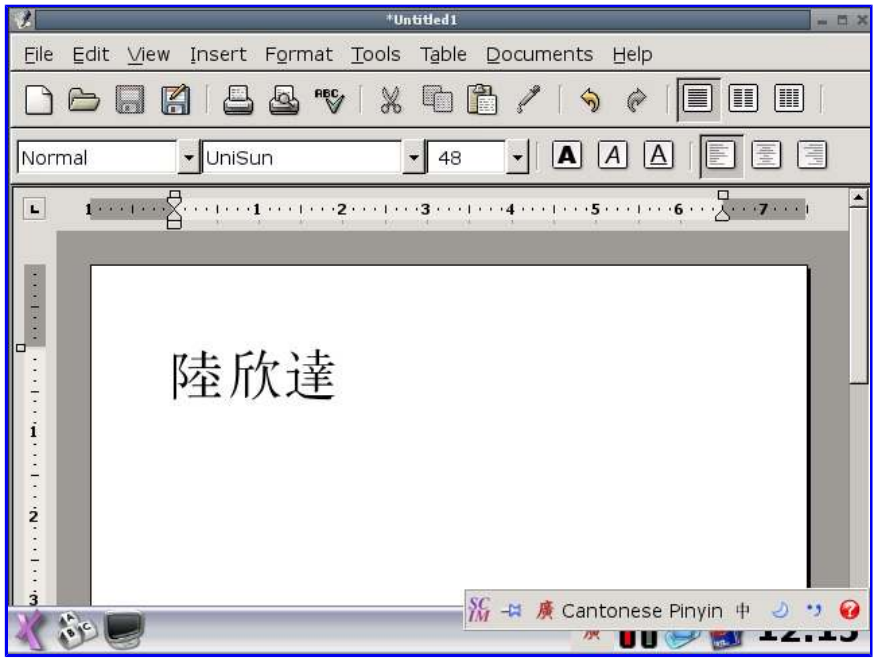
In addition, Wifi Radar is also pre-installed and configured for most devices and can be used to detect wireless networks.

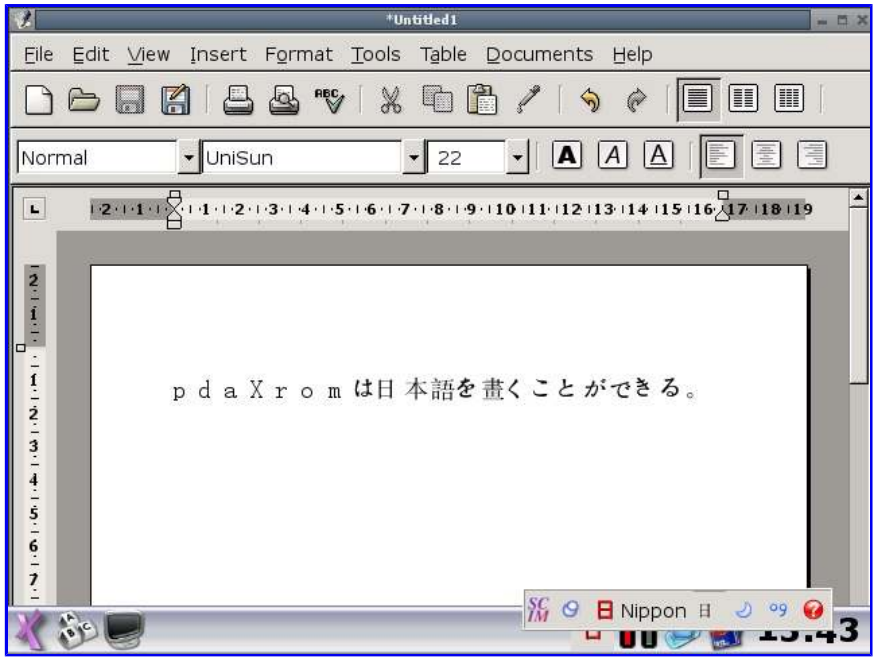
Chinese and Japanese support

Chinese and Japanese input can be enabled in pdaXii13 by invoking SCIM. You can configure SCIM by running **scim-setup**.



You can activate and switch input methods by pressing *Ctrl+space* for those applications that support it. AbiWord for example can be used with scim:





After installing the scim packages, you will also need to set the XMODIFIERS environment variable as follows:

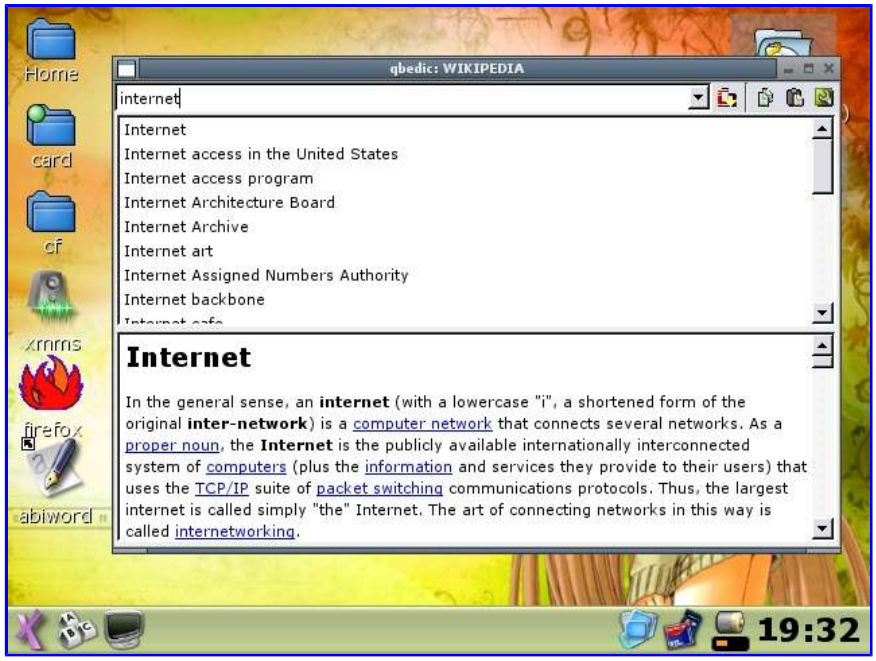
```
export XMODIFIERS=@im=SCIM
export GTK_IM_MODULE=scim
export QT_IM_MODULE=scim
```

Place the above into /etc/profile or .xinitrc

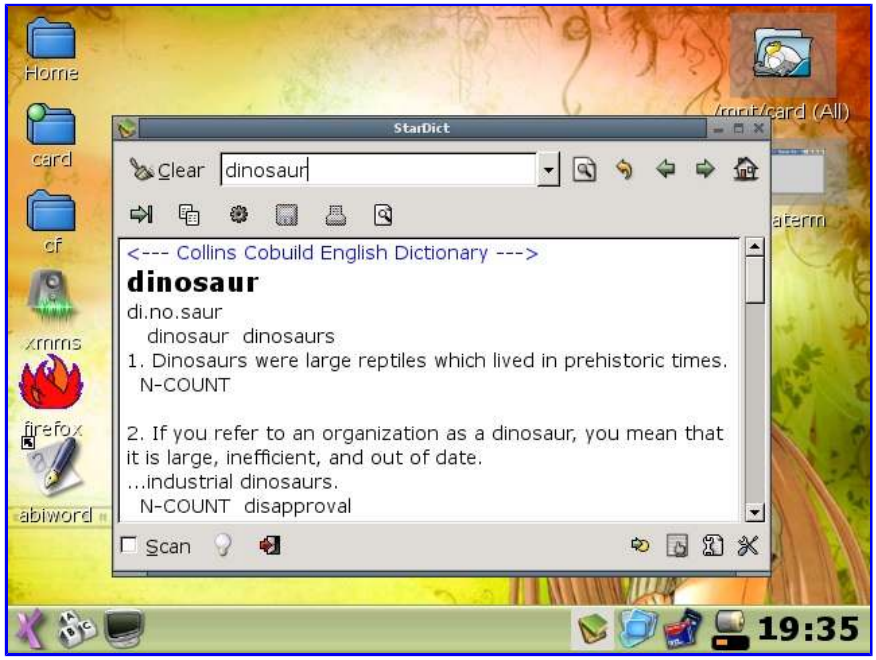
pdaXii13 has been preconfigured with scim-anthy and scim-pinyin to support Japanese and Chinese input.

Dictionaries

There are several dictionary packages available for pdaXrom. QBEdic is a popular dictionary between different languages that is also available for the Sharp ROM (called ZBEDict), so the same dictionary files can be used. There is also a complete wikipedia that can be installed into QBEDic for reference. QBEDic dictionary files can be placed into any directory, but by default they are located under /usr/share/bedic. You can configure QBEDic to look in specific directories for dictionary files.


Home
card
cf
xmms
firefox
abiword
qbedic: WIKIPEDIA
internet
Internet
Internet access in the United States
Internet access program
Internet Architecture Board
Internet Archive
Internet art
Internet Assigned Numbers Authority
Internet backbone
Internet
In the general sense, an internet (with a lowercase "i", a shortened form of the original inter-network) is a computer network that connects several networks. As a proper noun, the Internet is the publicly available internationally interconnected system of computers (plus the information and services they provide to their users) that uses the TCP/IP suite of packet switching communications protocols. Thus, the largest internet is called simply "the" Internet. The art of connecting networks in this way is called internetworking.
19:32

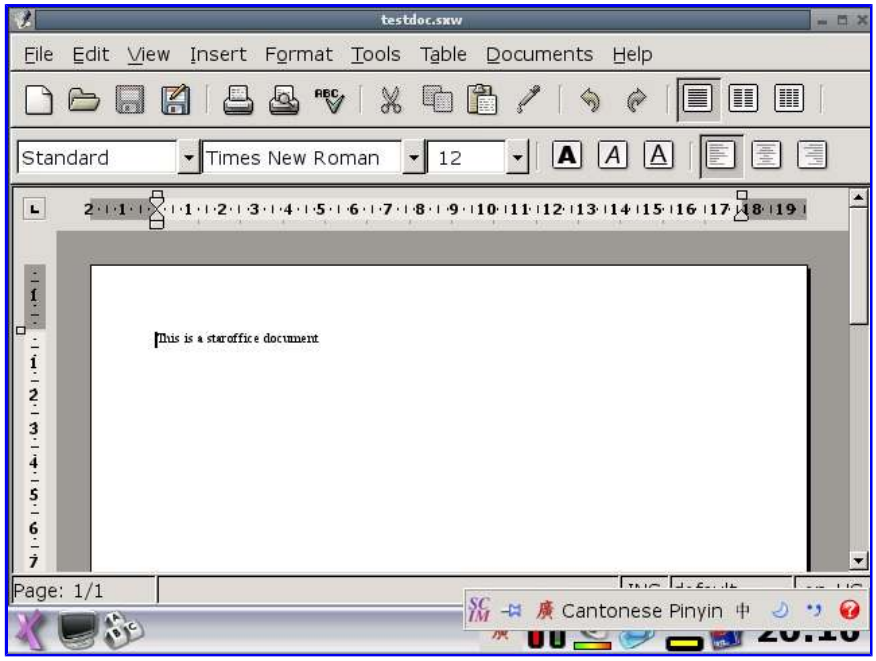
Another popular dictionary package for pdaXrom is stardict which also runs on Windows. There are many dictionary files available for stardict, and there is also a sound package for stardict to pronounce the words in English. Stardict expects its dictionary files to be located under /usr/share/stardict/dic. The sound files have to go under /usr/share



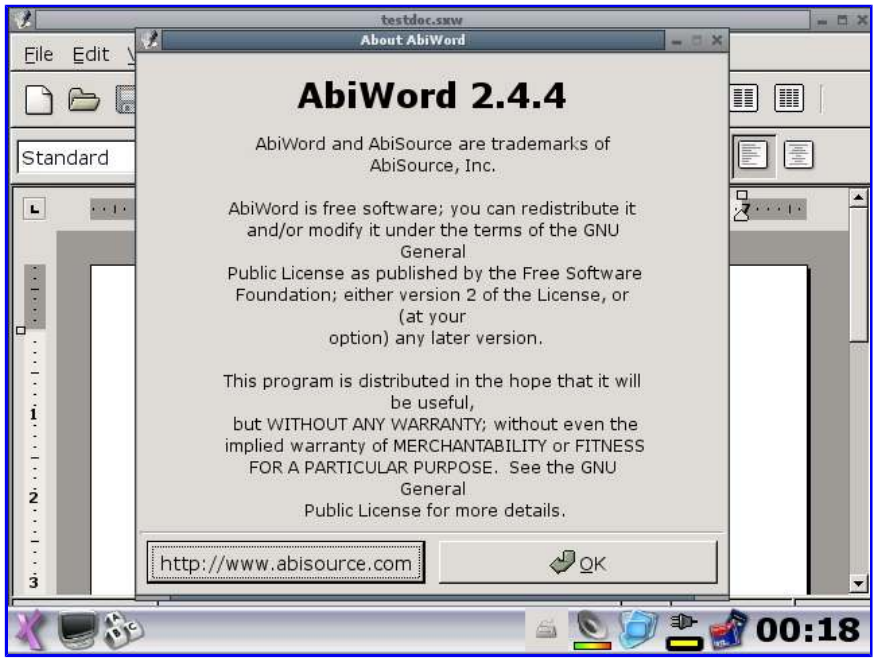
I have created a squashfs image containing several qbedic and stardict dictionaries as well as the sound files. Since the dictionary files are quite huge, placing them into a squashfs image can save quite a lot of space. I have also created a script called linkdics.sh which created the necessary links for stardict so that the files on the squashfs image can be located by stardict. Just place the squashfs image under /data and it will be automatically mounted when you reboot.

AbiWord

Abiword is a nice word processor, but it defaults to a 800x600 screen resolution which is not very usable for a Zaurus. For the C3000, it should be changed to 640x480. This can be done by editing */home/root/.AbiSuite/AbiWord.Profile*



I have compiled Abiword 2.4.4 and fixed all the window sizes so now every window in Abiword is sized correctly. I also compiled some text import/export plugins, so now I can open and edit StarOffice/OpenOffice documents.


testdoc.sxw
About AbiWord
AbiWord 2.4.4
AbiWord and AbiSource are trademarks of AbiSource, Inc.
AbiWord is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
http://www.abisource.com
OK
00:18

aspell is also installed to enable spell checking in abiword.

mplayer

The mplayer binary compiled by Agawa Koji (atty) is the most optimised and best performing mplayer binary available for the C3000 and thus is the default in pdaXii13.

The following mplayer command line will allow you to watch video:

```
# mplayer -vo bvdd -ac mad -ao ss -framedrop -really-quiet file
```

To force fullscreen playback, add the -vm option:

```
# mplayer -vo bvdd -ac mad -ao ss -framedrop -really-quiet -vm file
```

The xmmsmplayer addon to xmms can be used as a frontend for mplayer. Unfortunately, while mplayer is active and playing taking screenshots is not possible and thus the screenshot seen here has been modified and does not look so good.

It uses the -vo x11 option by default to play the video in a gtk window. You can make it use the bvdd driver instead and play fullscreen also. The config file used by the plugin is */usr/share/mplayer/mplayer.conf* and contains the following config:

```
framedrop = yes
cache = 1024
dr = yes
af=resample=44100
```

Add the following to make mplayer use the bvdd driver and play fullscreen:

```
vo = bvdd
vm = yes
```

There is a bug with mplayer that it does not refresh the screen properly once it exits from fullscreen mode. As a workaround, you can run **xrandr -o normal** to refresh the screen. Alternatively, you can simply press *Home+5* to refresh the screen.

I have also compiled the latest version of mplayer 1.0rc1 with gmplayer enabled. It runs slower but supports additional video and audio codecs.

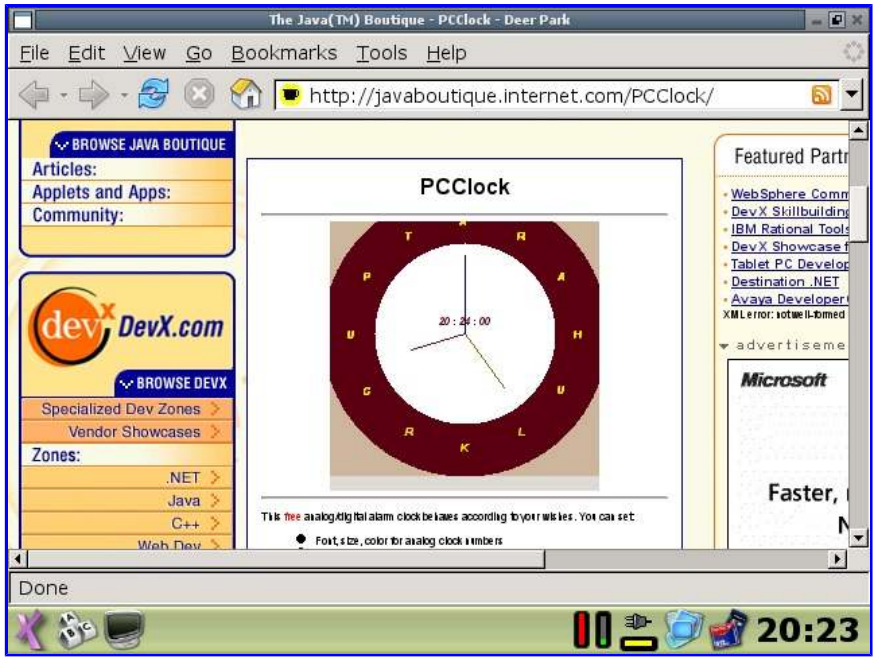
Alternatively, smplayer is a QT based mplayer frontend which I customised to be able to run atty's optimized mplayer-bvdd.

In pdaXii13, mplayer is a wrapper script which calls the real mplayer binaries (mplayer1, mplayer2, mplayer3) depending on the options set. bvdd is the default and xrandr is automatically called to refresh the screen after mplayer exits.

Java

Java is available through the jamvm and classpath packages and is roughly comparable to JRE 1.4.x but also implements some of the JRE 1.5.x features.

In addition, a Java Plugin for Firefox which uses jamvm and classpath is also available [gcjwebplugin-0.3.2_armv5tel.ipk] and enables many Java applets in Firefox 1.5, however, not all applets will work.



The jamvm and classpath packages gives you a Java compatable runtime, however, in order to develop Java applications, you will need a Java compiler and other tools like jar, javah, javap and javadoc. Jikes can be used as a replacement Java compiler, and there is also a classpath tools package which has some of the Java tools you need.

You can copy the tools.jar file from a 1.4.x JDK and use it instead.

gcc

There is an on-board gcc 3.4.6 compiler to allow native development on the Zaurus. The older 3.4.5 version is also compatible.

You need to put one of the zgcc images such as **zgcc-3.4.6.squashfs** or **zgcc-3.4.5-4.squashfs** under */home/root* or */data*. This will make it being automatically mounted on boot. You can also mount it manually as follows:

```
# mount -o loop /data/zgcc-3.4.6.squashfs /opt/native/arm/3.4.6-xscale-softvfp
```

To compile a little test application, create a file called hello.c with the following content:

```
#include <stdio.h>
int main()
{
printf("Hello World\n");
}
```

Then run the following command which will generate an executable called **hello**.

```
# gcc -o hello hello.c
```

To compile source packages, first download and extract the source tarball. Then do the following:

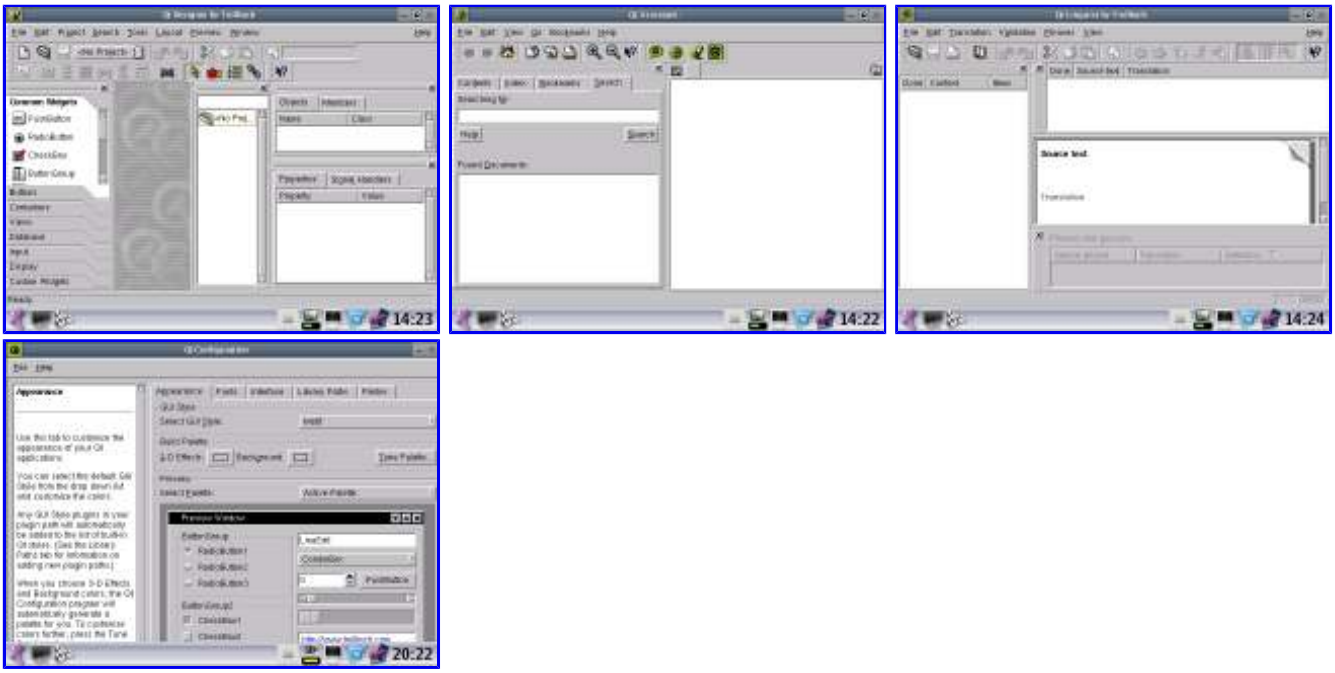
```
# ./configure
# ./make
```

You can also use **strip** to strip the binary to make it smaller once it is built. Usually, there is a DESTDIR variable defined in the Makefile which can help you turn the compiled binaries into a package.

```
# cd /data/build/pkg
# newipk testapp
# cd /data/build/src/testapp
# make DESTDIR=/data/build/pkg/testapp/data install
# cd /data/build/pkg
# makeipk testapp
```

This example uses ipk-tools to create a package. See the ipkg section below for further info.

You can use gcc 3.4.6 to compile both console and X based applications, however, you cannot use it to compile the kernel or glibc. You will need gcc 2.95-2 instead.



Furthermore, I also added QT designer, assistant and linguists, and I recompiled qtconfig so it won't startup oversized.

Building your own Packages

You can also build your own packages (ipk files) if you have written some useful scripts or written some applications that you want to distribute and let others install easily with the standard package manager.

I have build a package ipktools [ipktools_0.2_armv5tel.ipk] which has a set of tools for manipulating ipk files on pdaXrom:

- newipk - creates a package template structure for you to add files to for packaging
- makeipk - package up a directory that contains files in an ipk structure into an ipk file
- unpackipk - extracts the contents of an ipk file into a directory structure for repackaging
- deb2ipk - converts a deb file into an ipk file format (needs perl)
- zipipk - zip up an ipk file and remove ipk file afterwards
- ipkg-make-index - creates the Packages file for a feed

There currently are two ipk file formats. One uses the tar and gz format, wheras the other one uses a different binary format that is the same as the Debian .deb format. The Zaurus with default Sharp ROM (and Cacko) as well as pdaXrom uses the tar and gz format, which basically is a gzipped tarball (.tgz or .tar.gz) with a control structure and renamed to .ipk. If you extract this ipk file, you will find 3 files inside it - a text file called debian-binaries which just contains the string *2.0*, and two .tar.gz files called control and data. The control.tar.gz file contains a text file called control which has informaton about the package such as the Maintainer's name, dependencies, version, description, etc. There may also be some optional shell scripts for doing some pre and post configuration tasks during install and uninstall. Finally, the file data.tar.gz contains all the files and directory structure of the files for their destination location. The other ipk format is used by OpenZaurus and PocketWorkstation (Debian) and requires the **ar** command to extract the files.

To unpack an ipk file to see what is inside it, do the following:

```
# unpackipk somefile.ipk
```

To create your own ipk file, do the following to create the ipk file structure:

```
# newipk myproject
```

Then once you add your files in the correct locations and also update the control file with the information about your application, you can create your ipk file with the following command:

```
# makeipk myproject
```

If you want to convert a .deb or OZ ipk file so you can unpack it with **unpackipk**, then convert it using the **deb2ipk** perl script first, or ue the **ar** tool instead.

Building your own pdaXii13 image

You can also build your own customised pdaXii13 image.

The simplest way is to just backup your customised pdaXii13 system with **zbackup** and rename the backup tgz file to hdimage-custom.tgz and use it to flash your Zaurus with (in combination with the other install files).

The harder way is to rip the default initrd file for the C1000/C3100 and customise that image.

The following describes how the pdaXii13 binaries are built if you want to build it yourself from scratch:

updater.sh

Grab an updater.sh file (I used the one from pdaXrom C3000 beta2) and use the **endecsh** utility to decode it into a normal shell script and customise the script.

```
# endecsh -d updater.sh updater.txt
```

Then use **endecsh** to encode the updater.sh script.

```
# endecsh -e updater.txt updater.sh
```

You can compile **endecsh** yourself or extract it from the updater-tools.bin file.

updater-tools.bin

The updater-tools.bin file is a tar file. It contains utilities from the pdaXrom tools.tar file.

```
# tar xvf tools.tar
```

I removed the kernel images and nand patches from the extracted tar file and added the **pivot_root** binary and **init** script as well as other tools to the bin directory and re-archived it.

```
# tar cvf updater-tools.bin tools
```

zImage.bin

This is a kernel image. I just renamed the kernel from the C3000 beta2 to zImage-2.4.20.bin. This way, I can use almost any 2.4.20 kernel image for the C3000, by simply renaming the kernel image to zImage-2.4.20.bin and flashing it. I have found that the C3000 beta2 kernel works best and is most

stable. I have tried using the Tetsu special kernel image as well and it mostly works, but since the C3000 beta2 kernel works great, there is no point in using the Tetsu kernel.

initrd.bin

The initrd.bin file from the pdaXrom C3000 beta2 was a rip from an old OZ distro that used the 2.4.20 kernel. Hence the system contained in the initrd.bin file required a specifically compiled kernel for OZ and also had to be compatible with pdaXrom. All the mini system does is boot up and mount the microdrive and then pivot to it. I ripped the emergency boot system to built the initrd.bin. To do that, mount mtdblock1 and tar it up.

```
# mount /dev/mtdblock1 /mnt/test
# cd /mnt/test
# tar cvf initrd.tar *
# cd
# umount /mnt/test
```

Then extract the initrd.tar file and modify it. Most importantly is to place **pivot_root** and **init** into sbin directory. Then use the mkfs.jffs tool to generate a jffs2 image:

```
# tar xvf -C initrd
# mkfs.jffs2 -n -e 16 -o initrd.jffs2 initrd
```

Now the generated initrd.jffs2 file needs to be prepended with the Sharp initrd header which you can extract from the original initrd file:

```
# head -c 16 initrd.bin > initrdheader
# cat initrdheader initrd.jffs2 > initrd.bin
```

You can extract **head** and **mkfs.jffs2** from updater-tools.bin and make sure you use the **head** utility from the extracted updater-tools bin directory rather than the default **head** command.

Later I found that the initrd.bin from the C3000 beta2 was sufficient, so I ripped that one instead and customised it.

hdimage.tgz

The hdimage.tgz is simply a gzipped tarball from the pdaXrom beta3 initrd.bin file. You can customise the content of this file, ie add and remove applications to and from it and also customise the pre-configured settings. The easiest way is to flash pdaXii13, then customise the running system to your liking and then use **zbackup** to generate a tgz image of your current system. Rename the backup tgz image to hdimage-custom.tgz and the installer will use it to flash your C3000 instead of the default hdimage.tgz or hdimage-base.tgz. hdimage-base.tgz contains the files from the ripped C1000/C3100 initrd.bin with minimal customisations. hdimage-full.tgz contains an image of a more heavily customised version of the ripped initrd.bin file with many more pre-installed applications.

Note that the file structure between hdimage-base and hdimage-full differs slightly to the structure in hdimage-custom. Both hdimage-base and hdimage-full have all the files and directory structures located under /hdd1 wheras hdimage-custom contains no parent /hdd1 directory.

pdaXii13 build

pdaXii13-build.tgz is a complete build image for creating the pdaXii13 flash/install files as well as the initrd.bin image. This build image can be used to build pdaXii13 from the Z directly. It works if you have Sharp/Cacko/pdaXrom installed on your Zaurus.

pdaXii13-custom.tgz contains all the customisation files that were used to customise the initrd image from beta3 to create hdimage-base.tgz

In theory (this has not been tested), one could use pdaXrom builder to build a custom image for the C3100, but instead of generating an initrd.bin file, you could tar up the generated image instead and use the files from the pdaXii13-custom.tgz tarball to customise the image for the C3000.

The shell script **install-fixes-beta3.sh** contained in pdaXii13-custom.tgz will apply all the basic customisations and could also be used to update a C1000/C3100/C3200 beta3 install with fixes similar to pdaXii13 base. A readme file is included in the tarball which describes each file in the bundle.

The build files for pdaXii13 can all be found at tyrannozaurus pdaXii13 build section.

Resources:

http://ezaurus.com/lineup/sl/index.html - Zaurus product page

http://support.ezaurus.com/sl-3000/qa/ - Zaurus SL-C3000 FAQ in Japanese

http://trisoft.de/zaurus.htm - One of the better places for buying a Zaurus (they have an English manual too)

http://www.streamlinecpus.com - TRIsoft's partner in the US

http://www.pulster.de - Another place in Europe for purchasing a Zaurus

http://pricejapan.com - Cheapest place for a Zaurus (if support is not important to you)

http://www.wolf.ne.jp/syuhen/zaurus.html - Zaurus accessories

http://www.zaurusworld.ne.jp/ - Sharp SpaceTown

http://software.ezaurus.com/ - Zaurus Software (Japanese)

http://www.zaurususergroup.org/ - Zaurus user group and software repository

http://www.oesf.org/forums/ - Zaurus help forum in English

http://www.linuxontour.de/modules/newbb/ - Zaurus forum in German

http://forum.zaurusfr.org - Zaurus forum in French

http://externe.net/zaurus/forum/index.php - Zaurus resource page in English and French

http://ssh.idv.tw/bbs/viewforum.php?f=30 - Zaurus forum in Chinese (ROC)

http://www.pumb.org/forumdisplay.php?fid=15 - Zaurus forum in Chinese (HK)

http://forum.zaurus.cn/ - Zaurus forum in Chinese (China)

http://www.sluser.org/slwiki/ - Zaurus resource wiki in Japanese

http://www.ayati.com/kobako/c3000.htm - Zaurus info page in Japanese

http://www.killefiz.de/zaurus/ - software repository

http://www.elsix.org - software repository

http://www.pdaxrom.org - alternate distro and software repository

http://www.openzaurus.org - alternate distro and software repository

http://www.zaurusoft.com/ - software repository

http://zsi2.stonekeep.com/ - software repository

http://atty.skr.jp - mplayer and emulators

http://www.gnurou.org/software/zaurus/ - games

http://zaurus.spy.org/ - lots of Z resources

http://www.takei.gr.jp/zaurus/zbook.html - Zaurus Resources and Applications (in Japanese)

http://csx.jp/~zaurus/ - Zaurus Resources and Applications (in Japanese)

http://www.areanine.gr.jp/~nyano/ - Zaurus Resources and Applications (in Japanese)

http://tetsu.homelinux.org/zaurus/kernel/ - Tetsu's enhanced kernels

http://noz.ub32.org/zaurus/emu.html - emulators for Zaurus

http://www.cartel-securite.fr/pbiondi/zaurus/zethereal.html - Ethereal for Zaurus

http://www.robfisher.net/zaurus/freeciv.html - freeciv

http://undertow.2y.net/zaurus/bin/ - misc zaurus binaries

http://www.skill-engineering.com/zaurus/ - screensaver and other useful tools

http://www.zaurusthemes.org - Zaurus Themes and Wallpapers

http://home.mchsi.com/~cmisip/zaurus.htm - zaurus resource page (in English)

http://www.wbcd.com/computing/zaurus/ - Dictionary Resources

http://ebsnap.lkj.jp/zaurus/ - Dictionary Applications and Files

http://tbox.jpn.org/linuzau/ - keyhelper and other tools

http://pocketworkstation.org - Debian PocketWorkstation for the Zaurus

http://people.debian.org/~rene/openoffice.org/test/arm/ - OpenOffice for the Zaurus

http://www.tyrannozaurus.com - news about what is happening in the Zaurus world

http://www.thesounddesign.com/zaurus/beta3.htm - info about latest stable version of pdaXrom

---

Feeds:

Cacko/Sharp ROM compatible

http://www.zaurususergroup.org/feed/ - ZUG feed

http://zaurus.spy.org/feeds/docs-zaurus-com/ - another zaurus feed

http://web.mol.ru/~zaurus/feed/ - another zaurus feed with Cacko and X/Qt packages

http://bryandeluca.com/cacko/feed/ - Cacko feed

http://xqt.sourceforge.jp/feed-testing/ - X/Qt feed

http://xqt.sourceforge.jp/feed/ - X/Qt feed

http://www.pdaxrom.org/oldunstable/ - pdaXrom feed (old but works with X/Qt)

http://mail.pdaxrom.org/downloads/1.1.0beta1/XQt/feed/ - pdaXrom 1.1.0beta1 feed (compatible with X/Qt)

http://pdaxqtrom.thegrinder.ws/files/feed/ - pdaXqtrom feed (compatible with X/Qt)

http://opie.handhelds.org/feed/unsupported/sharprom/ - old opie feed

http://sharpromfeed.home.linuxtogo.org/feed/ - new Sharp compatible feed build using OE

pdaXrom 1.1 beta1/beta3 compatible

http://distro.ibiblio.org/pub/linux/distributions/pdaxrom/download/1.1.0beta3/Zaurus-Cxx00/feed/ - pdaXrom beta3 feed

http://distro.ibiblio.org/pub/linux/distributions/pdaxrom/download/1.1.0beta1/Zaurus-C1000-C3100/feed/ - pdaXrom beta1 Cxx00 feed

http://distro.ibiblio.org/pub/linux/distributions/pdaxrom/download/1.1.0beta1/Zaurus-7x0-860/feed/ - pdaXrom beta1 Cxx0 feed

http://distro.ibiblio.org/pub/linux/distributions/pdaxrom/download/contrib/ - various pdaXrom contrib feeds

http://www.tyrannozaurus.com/feed/beta3/feed/ - combined pdaXrom beta3 compatible feed

http://www.tyrannozaurus.com/feed/beta3/custom/ - my custom/updates feed

http://www.hermocom.com/feeds/pdaxrom1.1.0beta3/ - hmc feed

http://pdaxrom.sourceforge.jp/feed/ - Japanese support file feed for pdaXrom

http://www.pdaxrom.org/feed/ - pdaXrom feed (old feed)

http://www.pdaxrom.org/unstable/ - pdaXrom feed (contribs)

http://zaurus.spy.org/feeds/cacko/pdaXrom/1.1.0/ - mirror and backup of all the official pdaXrom release feeds

http://z.drun.net/packages/ - custom packages

http://matrixmen.free.fr/zaurus/pdaxrom/oesf/feed/ - new packages

OpenZaurus 3.4.5.1 compatible

http://mirror.hentges.net/www.openzaurus.org/3.5.4.1/feed/ - Hentges feed

---

DISCLAIMER: The information contained in this book is provided AS IS. No assurance is given to the accuracy of the information or instructions provided. You may use this as a guide but **do not blame me** if anything bad happens to your system or your data. Use anything described in this book at your own risk. I shall not be made responsible for anything you do.